Project: DRA FRONT END FILTER PROJECT

Title: Proof of Security (IId)

Ref: DS/FMU/FEF/013 *Issue: Revision* : *2.6* *Date:* 5 June 2016

Status: Approved *Type:* Specification

Keywords:

Author:

Name	*Location*	*Signature*	*Date*
G. M. Prout	WIN01		

Authorisation for Issue:

Name	*Function*	*Signature*	*Date*
R.B. Jones	HAT Manager		

Abstract: This document provides a formal proof for the second conjunct of the security property on the relationship between *hide* and *updateState* for the DRA front end filter project RSRE 1C/6130.

Distribution: HAT FEF File
Simon Wiseman

0 DOCUMENT CONTROL

0.1 Contents List

0.2 Document Cross References

[1] DS/FMU/017. *Secure Database Technical Proposal.* High Assurance Team, ICL Secure Systems, WIN01, 21st January 1992.

[2] DS/FMU/FEF/007. *Proof Strategy.* G.M. Prout, ICL Secure Systems, WIN01.

[3] DS/FMU/FEF/011. *Proof of Security (IIb).* G.M. Prout, ICL Secure Systems, WIN01.

[4] DS/FMU/FEF/012. *Proof of Security (IIc).* G.M. Prout, ICL Secure Systems, WIN01.

0.3 Changes History

Issue *Revision* : *2.6* **(5 June 2016)** Corrected LaTeX error.

Issue 2.6 Removed dependency on ICL logo font

Issue 2.7 Allowed for changes to forward chaining.

0.4 Changes Forecast

None.

1 GENERAL

1.1 Scope

This document, together with [3] and [4], provides a formal proof that the components *hide* and *updateState* satisfy their critical requirements, as specified in the proof strategy [2]. It constitutes part of deliverable D6 of work package 1c, as given in section 7 of the Secure Database Technical Proposal, [1].

1.2 Introduction

This document is a proof script which provides a formal proof which contributes to the proof of the second conjunct of *Lemma1*, the requirement on the critical components *hide* and *updateState*, described in the proof strategy document [2].

Lemma1

$$?\vdash \quad hide \in secureHide \land (hide, updateState) \in secureUpdate$$

In this document, we give a proof that the second conjunct of *secureUpdate* holds for *hideR* and *updateStateR*:

$$?\vdash \forall\ c_1\ c_2\ s_1\ s_2\ e$$
$$\bullet\ hideR\ (c_1,\ repState\ s_1) = hideR\ (c_1,\ repState\ s_2) \land c_1\ dominates\ c_2$$
$$\Rightarrow hideR\ (c_1,\ Fst\ (updateStateR\ (c_2,\ e,\ repState\ s_1)))$$
$$= hideR\ (c_1,\ Fst\ (updateStateR\ (c_2,\ e,\ repState\ s_2)))$$

2 PRELIMINARIES

The following ProofPower instructions set up the new theory *fef013*.

SML

```
open_theory "fef012";
(force_delete_theory "fef013" handle _ => ());
new_theory"fef013";
push_merge_pcs["hol","wrk049","wrk049a","'pair1"] ;
```

3 PROOF OF CONJUNCT 2

3.1 Insert Rows Lemma

```
SML
push_goal([],⌜∀ c1 c2 t1 t2 ds •
        cleanTable c1 t1 = cleanTable c1 t2 ∧ c1 dominates c2
        ⇒
        cleanTable c1 (replaceRows t1
                ((TS_rows t1) ⌢ (Map((MkRow c2) o (colDefaults c2 t1))ds)))
        =
        cleanTable c1 (replaceRows t2
                ((TS_rows t2) ⌢ (Map((MkRow c2) o (colDefaults c2 t2))ds)))⌝);
a(REPEAT strip_tac);
a(DROP_NTH_ASM_T 2 ante_tac THEN rewrite_tac
        [cleanTable_def,replaceRows_def,get_spec⌜MkTableSpec⌝]);
a(cases_tac⌜c1 dominates TS_class t1⌝THEN cases_tac⌜c1 dominates TS_class t2⌝
        THEN asm_rewrite_tac[]);
(* *** Goal "1" *** *)
a(rewrite_tac[get_spec⌜MkTableSpec⌝,tab_components,cleanColCons_def]
        THEN REPEAT strip_tac);
a(POP_ASM_T ante_tac THEN rewrite_tac[cleanRows_def,revealRow_def,map_⌢_thm1]);
a ⇒_tac;
```

SML

```
a(LEMMA_T`Map
                (cleanRow
                    c1
                    {col
                      |∃ y
                        • c1 dominates CC_exist y
                            ∧ (CS_consGroup col, y) ∈ TS_cons t1})
                (Map (MkRow c2 o colDefaults c2 t1) ds
                    ↾ {r|c1 dominates R_exist r}) =
        Map
                (cleanRow
                    c1
                    {col
                      |∃ y
                        • c1 dominates CC_exist y
                            ∧ (CS_consGroup col, y) ∈ TS_cons t2})
                (Map (MkRow c2 o colDefaults c2 t2) ds
                    ↾ {r|c1 dominates R_exist r})'asm_rewrite_thm_tac);
a(POP_ASM_T (fn _ => id_tac));
a(REV_LIST_INDUCTION_T`ds'asm_tac);
```

SML

```
(* *** Goal "1.1" *** *)
a(rewrite_tac[map_def,↾_def]);
(* *** Goal "1.2" *** *)
a(rewrite_tac[↾_thm,get_spec⌜MkRow⌝]);
a(∀_tac THEN GET_NTH_ASM_T 8 rewrite_thm_tac);
a(LEMMA_T⌜cleanRow  c1 {col
                  |∃ y
                    • c1 dominates CC_exist y
                        ∧ (CS_consGroup col, y) ∈ TS_cons t1}
                (MkRow c2 (colDefaults c2 t1 last))
              = cleanRow c1  {col
                  |∃ y
                    • c1 dominates CC_exist y
                        ∧ (CS_consGroup col, y) ∈ TS_cons t2}
                (MkRow c2 (colDefaults c2 t2 last))⌝asm_rewrite_thm_tac);
a(POP_ASM_T (fn _ => id_tac));
a(POP_ASM_T(strip_asm_tac o rewrite_rule[rel_ext_clauses]));
a(asm_rewrite_tac[colDefaults_def,cleanColCons_def,visibleCols_def,cleanRow_def,
        get_spec⌜MkRow⌝,row_components,rel_ext_clauses,filterRow_def]);
a(REPEAT strip_tac);
```

SML

```
(* *** Goal "1.2.1" *** *)
a(cases_tac⌜¬ x ∈ Dom{(n, d)
              |∃ c
               • ¬ (∃ y
                     • c₂ dominates CC_exist y
                         ∧ (CS_consGroup c, y) ∈ TS_cons t₁)
                   ∧ n = CS_posn c
                   ∧ d = CS_default c}⌝);
(* *** Goal "1.2.1.1" *** *)
a(DROP_NTH_ASM_T 3 ante_tac THEN asm_rewrite_tac[⊕_thm]);
a(rewrite_tac[⊕_thm] THEN strip_tac);
(* *** Goal "1.2.1.1.1" *** *)
a(∃_tac⌜z⌝THEN asm_rewrite_tac[] THEN strip_tac);
(* *** Goal "1.2.1.1.1.1" *** *)
a(∃_tac⌜c⌝THEN asm_rewrite_tac[]);
a(∃_tac⌜y'⌝THEN asm_rewrite_tac[]);
(* *** Goal "1.2.1.1.1.2" *** *)
a ∨_left_tac;
a(REPEAT strip_tac);
a(DROP_NTH_ASM_T 4(strip_asm_tac o rewrite_rule[dom_def]));
a(spec_nth_asm_tac 1 ⌜y''⌝);
a(spec_nth_asm_tac 1 ⌜c'⌝);
a(∃_tac⌜y'''⌝THEN asm_rewrite_tac[]);
a(lemma_tac⌜c₁ dominates CC_exist y'''⌝);
```

SML

```
(* *** Goal "1.2.1.1.1.2.1" *** *)
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.2.1.1.1.2.2" *** *)
a(list_spec_nth_asm_tac 13 [⌜CS_consGroup c'⌝,⌜y'''⌝]);
(* *** Goal "1.2.1.1.2" *** *)
a(DROP_NTH_ASM_T 4(strip_asm_tac o rewrite_rule[dom_def]));
a(spec_nth_asm_tac 1 ⌜z⌝);
a(spec_nth_asm_tac 1 ⌜c'⌝);
a(spec_nth_asm_tac 7 ⌜y''⌝);
(* *** Goal "1.2.1.2" *** *)
a(DROP_NTH_ASM_T 3 ante_tac THEN asm_rewrite_tac[⊕_thm]);
a(rewrite_tac[⊕_thm] THEN strip_tac);
a(∃_tac⌜z⌝THEN asm_rewrite_tac[] THEN strip_tac);
```

SML

```
(* *** Goal "1.2.1.2.1" *** *)
a(∃_tac⌜c⌝ THEN asm_rewrite_tac[]);
a(∃_tac⌜y'⌝ THEN asm_rewrite_tac[]);
(* *** Goal "1.2.1.2.2" *** *)
a ∨_right_tac;
a(∃_tac⌜c'⌝ THEN asm_rewrite_tac[]);
a(REPEAT strip_tac);
a(spec_nth_asm_tac 4 ⌜y''⌝);
a contr_tac;
a(lemma_tac⌜c1 dominates CC_exist y''⌝);
(* *** Goal "1.2.1.2.2.1" *** *)
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
(* *** Goal "1.2.1.2.2.2" *** *)
a(list_spec_nth_asm_tac 13 [⌜CS_consGroup c'⌝,⌜y''⌝]);
```

SML

```
(* *** Goal "1.2.2" *** *)
a(cases_tac⌜¬ x ∈ Dom{(n, d)
              |∃ c
                • ¬ (∃ y
                       • c2 dominates CC_exist y
                           ∧ (CS_consGroup c, y) ∈ TS_cons t2)
                   ∧ n = CS_posn c
                   ∧ d = CS_default c}⌝);
(* *** Goal "1.2.2.1" *** *)
a(DROP_NTH_ASM_T 3 ante_tac THEN asm_rewrite_tac[⊕_thm]);
a(rewrite_tac[⊕_thm] THEN strip_tac);
(* *** Goal "1.2.2.1.1" *** *)
a(∃_tac⌜z⌝ THEN asm_rewrite_tac[] THEN strip_tac);
(* *** Goal "1.2.2.1.1.1" *** *)
a(∃_tac⌜c⌝ THEN asm_rewrite_tac[]);
a(∃_tac⌜y'⌝ THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "1.2.2.1.1.2" *** *)
a ∨_left_tac;
a(REPEAT strip_tac);
a(DROP_NTH_ASM_T 4(strip_asm_tac o rewrite_rule[dom_def]));
a(spec_nth_asm_tac 1 ⌜y''⌝);
a(spec_nth_asm_tac 1 ⌜c'⌝);
a(∃_tac⌜y'''⌝THEN asm_rewrite_tac[]);
a(lemma_tac⌜c1 dominates CC_exist y'''⌝);
(* *** Goal "1.2.2.1.1.2.1" *** *)
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.2.2.1.1.2.2" *** *)
a(list_spec_nth_asm_tac 13 [⌜CS_consGroup c'⌝,⌜y'''⌝]);
(* *** Goal "1.2.2.1.2" *** *)
a(DROP_NTH_ASM_T 4(strip_asm_tac o rewrite_rule[dom_def]));
a(spec_nth_asm_tac 1 ⌜z⌝);
a(spec_nth_asm_tac 1 ⌜c'⌝);
a(spec_nth_asm_tac 7 ⌜y''⌝);
```

SML

```
(* *** Goal "1.2.2.2" *** *)
a(DROP_NTH_ASM_T 3 ante_tac THEN asm_rewrite_tac[⊕_thm]);
a(rewrite_tac[⊕_thm] THEN strip_tac);
a(∃_tac⌜z⌝THEN asm_rewrite_tac[] THEN strip_tac);
(* *** Goal "1.2.2.2.1" *** *)
a(∃_tac⌜c⌝THEN asm_rewrite_tac[]);
a(∃_tac⌜y'⌝THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2.2.2" *** *)
a ∨_right_tac;
a(∃_tac⌜c'⌝THEN asm_rewrite_tac[]);
a(REPEAT strip_tac);
a(spec_nth_asm_tac 4 ⌜y''⌝);
a contr_tac;
a(lemma_tac⌜c1 dominates CC_exist y''⌝);
(* *** Goal "1.2.2.2.2.1" *** *)
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.2.2.2.2.2" *** *)
a(list_spec_nth_asm_tac 13 [⌜CS_consGroup c'⌝,⌜y''⌝]);
```

SML

```
(* *** Goal "2" *** *)
a(rewrite_tac[get_spec⌜MkTableSpec⌝,tab_components,cleanColCons_def]
        THEN REPEAT strip_tac);
a(DROP_NTH_ASM_T 7 ante_tac THEN DROP_NTH_ASM_T 5 rewrite_thm_tac);
a(asm_rewrite_tac[]);
(* *** Goal "3" *** *)
a(rewrite_tac[get_spec⌜MkTableSpec⌝,tab_components,cleanColCons_def]
        THEN REPEAT strip_tac);
a(DROP_NTH_ASM_T 7 ante_tac THEN DROP_NTH_ASM_T 5 rewrite_thm_tac);
a(asm_rewrite_tac[]);
val cleanTable_insertRows_lemma = save_pop_thm"cleanTable_insertRows_lemma";
```

HOL output

```
cleanTable_insertRows_lemma =
⊢ ∀ c1 c2 t1 t2 ds
  • cleanTable c1 t1 = cleanTable c1 t2 ∧ c1 dominates c2
      ⇒ cleanTable
          c1
          (replaceRows
              t1
              (TS_rows t1 ⌢ Map (MkRow c2 o colDefaults c2 t1) ds))
        = cleanTable
          c1
          (replaceRows
              t2
              (TS_rows t2 ⌢ Map (MkRow c2 o colDefaults c2 t2) ds))
```

3.2 Delete Rows Lemmas

SML

```
push_goal([],⌜∀ c l x• Extract
                    (1 .. # (l ⌢ [x])
                       \ Squash (Id(Dom(ListRel (l  ⌢ [x]) ▷ {r|c dominates R_exist r})))
                          Image ns
                        ∩ {i|R_exist (Nth (l ⌢ [x]) i) = c}) (l ⌢ [x])
        =       if  # l + 1 ∈ (1 .. # (l ⌢ [x])
                       \ Squash (Id(Dom(ListRel (l  ⌢ [x]) ▷ {r|c dominates R_exist r})))
                          Image ns
                        ∩ {i|R_exist (Nth (l ⌢ [x]) i) = c})
                then
                (Extract
                     (1 .. # l
                       \ Squash (Id(Dom(ListRel l ▷ {r|c dominates R_exist r})))
                          Image ns
                        ∩ {i|R_exist (Nth l i) = c}) l) ⌢ [x]
                else
                (Extract
                     (1 .. # l
                       \ Squash (Id(Dom(ListRel l ▷ {r|c dominates R_exist r})))
                          Image ns
                        ∩ {i|R_exist (Nth l i) = c}) l)⌝);
```

SML

```
a(REPEAT strip_tac THEN rewrite_tac[extract_⌒_single_ax]);
a(cases_tac⌜c dominates R_exist x⌝THEN
        asm_rewrite_tac[squash_⌒_thm,image_∪_thm,∪_∩_thm]);
(* *** Goal "1" *** *)
a(CASES_T⌜ # l + 1
                  ∈ 1 .. # (l ⌒ [x])
                    \ (Squash
                          (Id (Dom (ListRel l ▷ {r|c dominates R_exist r})))
                       Image ns
                      ∩ {i|R_exist (Nth (l ⌒ [x]) i) = c}
                     ∪ {(#
                                 (Squash
                                    (Id
                                       (Dom
                                          (ListRel l
                                             ▷ {r
                                             |c
                                                 dominates R_exist
                                               r})))))
                                + 1, # l + 1)}
                       Image ns
                      ∩ {i|R_exist (Nth (l ⌒ [x]) i) = c})⌝rewrite_thm_tac);
```

SML

```
(* *** Goal "1.1 & 1.2"*** *)
a(rewrite_tac[extract_def]);
a(LEMMA_T⌜(1 .. # (l ⌢ [x])
                        \ (Squash
                              (Id
                                  (Dom
                                      (ListRel l
                                          ▷ {r|c dominates R_exist r})))
                            Image ns
                          ∩ {i|R_exist (Nth (l ⌢ [x]) i) = c}
                        ∪ {(#
                                  (Squash
                                      (Id
                                          (Dom
                                              (ListRel l
                                                  ▷ {r
                                                  |c
                                                      dominates R_exist
          r})))))
                                    + 1, # l + 1)}
                            Image ns
                          ∩ {i|R_exist (Nth (l ⌢ [x]) i) = c})
                    ◁ ListRel l) = (1 .. # l
                     \ Squash
                         (Id
                             (Dom
                                 (ListRel l
                                     ▷ {r|c dominates R_exist r})))
                           Image ns
                         ∩ {i|R_exist (Nth l i) = c}
                    ◁ ListRel l)⌝rewrite_thm_tac);
```

SML

```
a(rewrite_tac[rel_ext_clauses,◁_def,▷_def,list_rel_def,image_def,dot_dot_def,
        ≤_plus1_thm,length_⌢_one_thm]);
a(REPEAT strip_tac);
(* *** Goal "1.1.1" *** *)
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.1.2" *** *)
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.1.3" *** *)
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.1.4" *** *)
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.1.5" *** *)
a(spec_nth_asm_tac 5 ⌜x''⌝);
(* *** Goal "1.1.6" *** *)
a(spec_nth_asm_tac 5 ⌜x''⌝);
```

SML

```
(* *** Goal "1.1.7" *** *)
a(strip_asm_tac(rewrite_rule[dot_dot_def](list_∀_elim[⌜l⌝,⌜x'⌝,⌜x⌝]nth_⌢_thm1)));
a(DROP_NTH_ASM_T 6 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.1.8" *** *)
a(strip_asm_tac(rewrite_rule[dot_dot_def](list_∀_elim[⌜l⌝,⌜x'⌝,⌜x⌝]nth_⌢_thm1)));
a(DROP_NTH_ASM_T 5 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.1.9" *** *)
a(spec_nth_asm_tac 4 ⌜x''⌝);
(* *** Goal "1.1.10" *** *)
a(DROP_NTH_ASM_T 6 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.1.11" *** *)
a(strip_asm_tac(rewrite_rule[dot_dot_def](list_∀_elim[⌜l⌝,⌜x'⌝,⌜x⌝]nth_⌢_thm1)));
a(DROP_NTH_ASM_T 3 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.1.12" *** *)
a(DROP_NTH_ASM_T 6 ante_tac THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "2" *** *)
a(CASES_T⌜ # l + 1
                ∈ 1 .. # (l ⌢ [x])
                  \ Squash
                        (Id (Dom (ListRel l ▷ {r|c dominates R_exist r})))
                      Image ns
                    ∩ {i|R_exist (Nth (l ⌢ [x]) i) = c}⌝ rewrite_thm_tac);
```

SML

```
(* *** Goal "2.1 & 2.2" *** *)
a(rewrite_tac[extract_def]);
a(LEMMA_T⌜(1 .. # (l ⌢ [x])
                        \ Squash
                            (Id
                                (Dom
                                    (ListRel l
                                        ▷ {r|c dominates R_exist r})))
                            Image ns
                          ∩ {i|R_exist (Nth (l ⌢ [x]) i) = c}
                        ◁ ListRel l)=(1 .. # l
                         \ Squash
                            (Id
                                (Dom
                                    (ListRel l
                                        ▷ {r|c dominates R_exist r})))
                            Image ns
                          ∩ {i|R_exist (Nth l i) = c}
                        ◁ ListRel l)⌝rewrite_thm_tac);
```

SML

```
a(rewrite_tac[rel_ext_clauses,◁_def,▷_def,list_rel_def,image_def,dot_dot_def,
        ≤_plus1_thm,length_⌢_one_thm]);
a(REPEAT strip_tac);
(* *** Goal "2.1.1" *** *)
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "2.1.2" *** *)
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "2.1.3" *** *)
a(spec_nth_asm_tac 4 ⌜x''⌝);
(* *** Goal "2.1.4" *** *)
a(strip_asm_tac(rewrite_rule[dot_dot_def](list_∀_elim[⌜l⌝,⌜x'⌝,⌜x⌝]nth_⌢_thm1)));
a(DROP_NTH_ASM_T 5 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "2.1.5" *** *)
a(spec_nth_asm_tac 4 ⌜x''⌝);
(* *** Goal "2.1.6" *** *)
a(strip_asm_tac(rewrite_rule[dot_dot_def](list_∀_elim[⌜l⌝,⌜x'⌝,⌜x⌝]nth_⌢_thm1)));
a(DROP_NTH_ASM_T 3 ante_tac THEN asm_rewrite_tac[]);
val extract_⌢_single_lemma = save_pop_thm"extract_⌢_single_lemma";
```

HOL output

```
extract_⌢_single_lemma =
⊢ ∀ c l x
  • Extract
      (1 .. # (l ⌢ [x])
         \ Squash
             (Id (Dom (ListRel (l ⌢ [x]) ▷ {r|c dominates R_exist r})))
           Image ns
          ∩ {i|R_exist (Nth (l ⌢ [x]) i) = c})
      (l ⌢ [x])
    = (if
       # l + 1
         ∈ 1 .. # (l ⌢ [x])
           \ Squash
               (Id
                  (Dom
                     (ListRel (l ⌢ [x]) ▷ {r|c dominates R_exist r})))
             Image ns
            ∩ {i|R_exist (Nth (l ⌢ [x]) i) = c}
      then
       Extract
         (1 .. # l
            \ Squash (Id (Dom (ListRel l ▷ {r|c dominates R_exist r})))
              Image ns
             ∩ {i|R_exist (Nth l i) = c})
         l
        ⌢ [x]
      else
       Extract
        (1 .. # l
           \ Squash (Id (Dom (ListRel l ▷ {r|c dominates R_exist r})))
             Image ns
            ∩ {i|R_exist (Nth l i) = c})
        l)
```

SML

```
push_goal([],⌜∀ l1 l2 c1 c2 s •
        c1 dominates c2
        ∧
        Map(cleanRow c1 s)(l1 ↾ {r|c1 dominates R_exist r})
        =
        Map(cleanRow c1 s)(l2 ↾ {r|c1 dominates R_exist r})
        ⇒
                Map(cleanRow c2 s)(l1 ↾ {r|c2 dominates R_exist r})
                =
                Map(cleanRow c2 s)(l2 ↾ {r|c2 dominates R_exist r})⌝);
```

SML

```
a(REPEAT strip_tac);
a(POP_ASM_T ante_tac THEN intro_∀_tac(⌜l1⌝,⌜l1⌝));
a(LIST_INDUCTION_T⌜l2⌝asm_tac);
(* *** Goal "1" *** *)
a(rewrite_tac[map_def,all_∀_intro(eq_sym_rule(all_∀_elim list_rel_list_thm))]);
a(rewrite_tac[rel_ext_clauses]THEN REPEAT strip_tac);
a contr_tac;
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
(* *** Goal "2" *** *)
a(REPEAT ∀_tac);
a(intro_∀_tac(⌜x⌝,⌜x⌝));
a(LIST_INDUCTION_T⌜l1⌝asm_tac);
a(rewrite_tac[map_def]);
```

SML

```
(* *** Goal "2.1" *** *)
a(rewrite_tac[map_def,all_∀_intro(eq_sym_rule(all_∀_elim list_rel_list_thm))]);
a(rewrite_tac[rel_ext_clauses]THEN REPEAT strip_tac);
a contr_tac;
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
(* *** Goal "2.2" *** *)
a(rewrite_tac[↑_def]);
a(REPEAT ∀_tac);
a(cases_tac⌜c2 dominates R_exist x⌝THEN cases_tac⌜c2 dominates R_exist x'⌝
        THEN asm_rewrite_tac[map_def]);
(* *** Goal "2.2.1" *** *)
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
a(asm_rewrite_tac[map_def]THEN strip_tac);
a(DROP_NTH_ASM_T 11 (ante_tac o ∀_elim⌜l1⌝)THEN asm_rewrite_tac[]
        THEN strip_tac THEN asm_rewrite_tac[]);
a(DROP_NTH_ASM_T 3 ante_tac THEN rewrite_tac
        [cleanRow_def,row_components,get_spec⌜MkRow⌝]);
a(rewrite_tac[rel_ext_clauses]THEN REPEAT strip_tac);
```

SML

```
(* *** Goal "2.2.1.1" *** *)
a(DROP_NTH_ASM_T 3 (asm_tac o list_∀_elim[⌜x''⌝,⌜replaceData c1 z⌝]));
a(LEMMA_T⌜(∃ z'
          • (x'', z') ∈ filterRow s (R_data x)
            ∧ replaceData c1 z = replaceData c1 z')⌝asm_tac);
(* *** Goal "2.2.1.1.1" *** *)
a(∃_tac⌜z⌝ THEN asm_rewrite_tac[]);
(* *** Goal "2.2.1.1.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(∃_tac⌜z'⌝ THEN asm_rewrite_tac[]);
a(fc_tac[replaceData_lemma] THEN asm_fc_tac[]);
(* *** Goal "2.2.1.2" *** *)
a(DROP_NTH_ASM_T 3 (asm_tac o list_∀_elim[⌜x''⌝,⌜replaceData c1 z⌝]));
a(LEMMA_T⌜(∃ z'
          • (x'', z') ∈ filterRow s (R_data x')
            ∧ replaceData c1 z = replaceData c1 z')⌝asm_tac);
(* *** Goal "2.2.1.2.1" *** *)
a(∃_tac⌜z⌝ THEN asm_rewrite_tac[]);
(* *** Goal "2.2.1.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(∃_tac⌜z'⌝ THEN asm_rewrite_tac[]);
a(fc_tac[replaceData_lemma] THEN asm_fc_tac[]);
```

SML

```
(* *** Goal "2.2.2" *** *)
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
a(cases_tac⌜c1 dominates R_exist x'⌝THEN asm_rewrite_tac[map_def,↑_def]);
(* *** Goal "2.2.2.1" *** *)
a strip_tac;
a(DROP_NTH_ASM_T 2
        (ante_tac o rewrite_rule[cleanRow_def,row_components,get_spec⌜MkRow⌝]));
a strip_tac;
a(DROP_NTH_ASM_T 9 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "2.2.2.2" *** *)
a strip_tac;
a(DROP_NTH_ASM_T 9(ante_tac o ∀_elim⌜Cons x l1⌝));
a(asm_rewrite_tac[↑_def,map_def]);
```

SML

```
(* *** Goal "2.2.3" *** *)
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
a(cases_tac⌜c1 dominates R_exist x⌝THEN asm_rewrite_tac[map_def,↑_def]);
(* *** Goal "2.2.3.1" *** *)
a strip_tac;
a(DROP_NTH_ASM_T 2
        (ante_tac o rewrite_rule[cleanRow_def,row_components,get_spec⌜MkRow⌝]));
a strip_tac;
a(DROP_NTH_ASM_T 9 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "2.2.3.2" *** *)
a strip_tac;
a(DROP_NTH_ASM_T 8(ante_tac o ∀_elim⌜x'⌝));
a(asm_rewrite_tac[↑_def,map_def]);
```

SML

```
(* *** Goal "2.2.4" *** *)
a(cases_tac ⌜c1 dominates R_exist x⌝THEN cases_tac ⌜c1 dominates R_exist x'⌝
        THEN asm_rewrite_tac[map_def,↑_def]);
(* *** Goal "2.2.4.1" *** *)
a(strip_tac THEN DROP_NTH_ASM_T 8(ante_tac o ∀_elim⌜l1⌝)THEN asm_rewrite_tac[]);
(* *** Goal "2.2.4.2" *** *)
a(strip_tac THEN DROP_NTH_ASM_T 7(ante_tac o ∀_elim⌜Cons x l1⌝)
        THEN asm_rewrite_tac[map_def,↑_def]);
(* *** Goal "2.2.4.3" *** *)
a(strip_tac THEN DROP_NTH_ASM_T 6(ante_tac o ∀_elim⌜x'⌝)
        THEN asm_rewrite_tac[map_def,↑_def]);
val map_cleanRow_lemma1 = save_pop_thm"map_cleanRow_lemma1";
```

HOL output

```
map_cleanRow_lemma1 =
⊢ ∀ l1 l2 c1 c2 s
  • c1 dominates c2
        ∧ Map (cleanRow c1 s) (l1 ↑ {r|c1 dominates R_exist r})
          = Map (cleanRow c1 s) (l2 ↑ {r|c1 dominates R_exist r})
      ⇒ Map (cleanRow c2 s) (l1 ↑ {r|c2 dominates R_exist r})
        = Map (cleanRow c2 s) (l2 ↑ {r|c2 dominates R_exist r})
```

SML

```
push_goal([],⌜∀ l1 l2 c1 c2 s •
        c1 dominates c2
        ∧
        Map(cleanRow c1 s)(l1 ↾ {r|c1 dominates R_exist r})
        =
        Map(cleanRow c1 s)(l2 ↾ {r|c1 dominates R_exist r})
        ⇒
                # (ListRel l1 ▷ {r|c2 dominates R_exist r})
                =
                # (ListRel l2 ▷ {r|c2 dominates R_exist r})⌝);
```

SML

```
a(REPEAT strip_tac);
a(fc_tac[map_cleanRow_lemma1]THEN asm_fc_tac[]);
a(DROP_NTH_ASM_T 2 (fn _ => id_tac)THEN DROP_NTH_ASM_T 2 (fn _ => id_tac));
a(POP_ASM_T ante_tac);
a(intro_∀_tac(⌜l2⌝,⌜l2⌝));
a(REV_LIST_INDUCTION_T⌜l1⌝asm_tac);
(* *** Goal "1" *** *)
a(rewrite_tac[map_def]);
a(∀_tac THEN ⇒_T (asm_tac o rewrite_rule
        [all_∀_intro(eq_sym_rule(all_∀_elim list_rel_list_thm))]));
a(asm_rewrite_tac[]);
```

SML

```
(* *** Goal "2" *** *)
a(REPEAT ∀_tac);
a(intro_∀_tac(⌜last⌝,⌜last⌝));
a(REV_LIST_INDUCTION_T⌜l2⌝asm_tac);
(* *** Goal "2.1" *** *)
a(rewrite_tac[map_def]);
a(∀_tac THEN cases_tac⌜c2 dominates R_exist last⌝ THEN  asm_rewrite_tac[]);
a(⇒_T (asm_tac o rewrite_rule[all_∀_intro(eq_sym_rule(all_∀_elim list_rel_list_thm))]));
a(asm_rewrite_tac[list_rel_⌢_singleton_thm]);
```

SML

```
(* *** Goal "2.2" *** *)
a(REPEAT ∀_tac);
a(cases_tac⌜c2 dominates R_exist last'⌝
        THEN cases_tac⌜c2 dominates R_exist last⌝ THEN asm_rewrite_tac[]);
(* *** Goal "2.2.1" *** *)
a(⇒_tac THEN DROP_NTH_ASM_T 6 (ante_tac o ∀_elim⌜l2⌝) THEN asm_rewrite_tac[]
        THEN ⇒_tac);
a(asm_rewrite_tac[list_rel_⌢_singleton_thm]);
a(lemma_tac⌜∀ l last • (ListRel l ▷ {r|c2 dominates R_exist r}) ∩ {(# l + 1, last)}
        = {}⌝);
(* *** Goal "2.2.1.1" *** *)
a(REPEAT ∀_tac THEN rewrite_tac[list_rel_def,▷_def,dot_dot_def,rel_ext_clauses]);
a(REPEAT strip_tac);
a(DROP_NTH_ASM_T 2 ante_tac THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "2.2.1.2" *** *)
a(TOP_ASM_T (asm_tac o list_∀_elim[⌜l1⌝,⌜last'⌝]));
a(asm_tac(list_∀_elim[⌜l1⌝,⌜{r|c2 dominates R_exist r}⌝]fin_list_rel_▷_thm));
a(asm_tac(∀_elim⌜(# l1 + 1, last')⌝fin_set_thm5));
a(ante_tac(list_∀_elim[⌜ListRel l1 ▷ {r|c2 dominates R_exist r}⌝,
        ⌜{(# l1 + 1, last')}⌝]size_thm7)THEN asm_rewrite_tac[size_thm1,size_singleton_thm]);
a(⇒_T rewrite_thm_tac);
a(DROP_NTH_ASM_T 4 (asm_tac o list_∀_elim[⌜l2⌝,⌜last⌝]));
a(asm_tac(list_∀_elim[⌜l2⌝,⌜{r|c2 dominates R_exist r}⌝]fin_list_rel_▷_thm));
a(asm_tac(∀_elim⌜(# l2 + 1, last)⌝fin_set_thm5));
a(ante_tac(list_∀_elim[⌜ListRel l2 ▷ {r|c2 dominates R_exist r}⌝,
        ⌜{(# l2 + 1, last)}⌝]size_thm7)THEN asm_rewrite_tac[size_thm1,size_singleton_thm]);
a(⇒_T asm_rewrite_thm_tac);
```

SML

```
(* *** Goal "2.2.2" *** *)
a(DROP_NTH_ASM_T 3(ante_tac o ∀_elim⌜last'⌝)THEN asm_rewrite_tac[]);
a(⇒_T asm_tac THEN ⇒_tac);
a(DROP_NTH_ASM_T 2 ante_tac THEN asm_rewrite_tac[]THEN strip_tac);
a(asm_rewrite_tac[list_rel_⌒_▷_thm]);
(* *** Goal "2.2.3" *** *)
a(DROP_NTH_ASM_T 4(ante_tac o ∀_elim⌜l2 ⌒ [last]⌝) THEN asm_rewrite_tac[]);
a(⇒_T asm_tac THEN ⇒_tac);
a(DROP_NTH_ASM_T 2 ante_tac THEN asm_rewrite_tac[]THEN strip_tac);
a(asm_rewrite_tac[list_rel_⌒_▷_thm]);
(* *** Goal "2.2.4" *** *)
a(asm_rewrite_tac[list_rel_⌒_▷_thm]);
val map_cleanRow_lemma2 = save_pop_thm"map_cleanRow_lemma2";
```

HOL output

```
map_cleanRow_lemma2 =
⊢ ∀ l1 l2 c1 c2 s
  • c1 dominates c2
        ∧ Map (cleanRow c1 s) (l1 ↾ {r|c1 dominates R_exist r})
          = Map (cleanRow c1 s) (l2 ↾ {r|c1 dominates R_exist r})
      ⇒ # (ListRel l1 ▷ {r|c2 dominates R_exist r})
        = # (ListRel l2 ▷ {r|c2 dominates R_exist r})
```

SML

```
push_goal([],⌜∀ c1 c2 t1 t2 ns •
        cleanTable c1 t1 = cleanTable c1 t2
        ∧ c1 dominates c2 ∧ c2 dominates TS_class t1
        ⇒
        cleanTable c1 (replaceRows t1
                (Extract (1 .. #(TS_rows t1)) \
        ((revealRow c2 t1) Image ns ∩ {i|R_exist(Nth(TS_rows t1)i) = c2}))
        (TS_rows t1)))
        =
        cleanTable c1 (replaceRows t2
                (Extract (1 .. #(TS_rows t2)) \
        ((revealRow c2 t2) Image ns ∩ {i|R_exist(Nth(TS_rows t2)i) = c2}))
        (TS_rows t2)))⌝);
a(REPEAT strip_tac);
a(lemma_tac⌜cleanTable c2 t1 = cleanTable c2 t2⌝
        THEN_LIST[fc_tac[cleanTable_lemma]
        THEN asm_fc_tac[],id_tac]);
a(DROP_NTH_ASM_T 4 ante_tac THEN
        rewrite_tac[cleanTable_def,replaceRows_def,get_spec⌜MkTableSpec⌝]);
a(lemma_tac⌜c1 dominates TS_class t1⌝THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(cases_tac⌜c1 dominates TS_class t2⌝THEN asm_rewrite_tac[]);
```

SML

```
set_labelled_goal"2";
(* *** Goal "2" *** *)
a(rewrite_tac[get_spec⌜MkTableSpec⌝,tab_components,cleanColCons_def]
        THEN REPEAT strip_tac);
a(DROP_NTH_ASM_T 7 ante_tac THEN DROP_NTH_ASM_T 5 rewrite_thm_tac);
a(asm_rewrite_tac[]);
```

SML

```
(* *** Goal "1" *** *)
a(⇒_T(strip_asm_tac o rewrite_rule[get_spec⌜MkTableSpec⌝,tab_components]));
a(DROP_NTH_ASM_T 8 ante_tac THEN asm_rewrite_tac[cleanTable_def]);
a(GET_NTH_ASM_T 8 ante_tac THEN GET_NTH_ASM_T 5 rewrite_thm_tac
        THEN ⇒_tac
        THEN asm_rewrite_tac[get_spec⌜MkTableSpec⌝,tab_components] THEN strip_tac);
a(POP_ASM_T ante_tac THEN lemma_tac⌜cleanColCons c2 t1 = cleanColCons c2 t2⌝
        THEN_LIST[pure_once_asm_rewrite_tac[prove_rule[pair_clauses]
        ⌜∀ p • p = (Fst p, Snd p)⌝]THEN asm_rewrite_tac[],
        DROP_NTH_ASM_T 3(fn _ => id_tac)
        THEN  DROP_NTH_ASM_T 2(fn _ => id_tac)]);
a(asm_rewrite_tac[] THEN ⇒_tac);
a(lemma_tac⌜cleanColCons c1 t1 = cleanColCons c1 t2⌝
        THEN_LIST[pure_once_asm_rewrite_tac[prove_rule[pair_clauses]
        ⌜∀ p • p = (Fst p, Snd p)⌝]THEN asm_rewrite_tac[],
        DROP_NTH_ASM_T 7(fn _ => id_tac)
        THEN  DROP_NTH_ASM_T 6(fn _ => id_tac)]);
a(DROP_NTH_ASM_T 5 ante_tac THEN asm_rewrite_tac[] THEN ⇒_tac);
```

SML

```
a(fc_tac[cleanRows_size_lemma]);
a(LIST_DROP_NTH_ASM_T [1,2,3,4](MAP_EVERY ante_tac));
a(LIST_DROP_NTH_ASM_T [1,2](MAP_EVERY (fn _ => id_tac)));
a(rewrite_tac[cleanColCons_def,cleanRows_def,revealRow_def,get_spec⌜MkTableSpec⌝]);
a(REPEAT strip_tac);
a(LIST_DROP_NTH_ASM_T [1,2,3](MAP_EVERY ante_tac));
a(lemma_tac⌜∃ l1 • TS_rows t1 = l1⌝THEN_LIST
        [prove_∃_tac,POP_ASM_T rewrite_thm_tac]);
a(lemma_tac⌜∃ l2 • TS_rows t2 = l2⌝THEN_LIST
        [prove_∃_tac,POP_ASM_T rewrite_thm_tac]);
a(intro_∀_tac(⌜l2⌝,⌜l2⌝));
a(REV_LIST_INDUCTION_T⌜l1⌝asm_tac);
```

SML

```
(* *** Goal "1.1" *** *)
a(rewrite_tac[map_def]THEN REPEAT strip_tac);
a(fc_tac[↑_extract_null_thm]);
a(spec_nth_asm_tac 1⌜(1 .. # l2
                        \ Squash
                            (Id
                                (Dom
                                    (ListRel l2
                                        ▷ {r|c2 dominates R_exist r})))
                          Image ns
                         ∩ {i|R_exist (Nth l2 i) = c2})⌝);
a(asm_rewrite_tac[extract_def,rel_list_null_thm]);
(* *** Goal "1.2" *** *)
a(REPEAT ∀_tac);
a(intro_∀_tac(⌜last⌝,⌜last⌝) THEN REV_LIST_INDUCTION_T⌜l2⌝asm_tac);
```

SML

```
(* *** Goal "1.2.1" *** *)
a(REPEAT ∀_tac THEN cases_tac⌜c1 dominates R_exist last⌝THEN asm_rewrite_tac
      [map_def,extract_⌢_single_ax] THEN REPEAT strip_tac);
a(CASES_T⌜# l1 + 1
                      ∈ 1 .. # (l1 ⌢ [last])
                        \ Squash
                            (Id
                                (Dom
                                    (ListRel (l1 ⌢ [last])
                                        ▷ {r|c2 dominates R_exist r})))
                          Image ns
                         ∩ {i|R_exist (Nth (l1 ⌢ [last]) i) = c2}⌝rewrite_thm_tac);
```

SML

```
(* *** Goal "1.2.1.1" *** *)
a(fc_tac[↑_extract_null_thm]);
a(spec_nth_asm_tac 1⌜(1 .. # (l1 ⌢ [last])
                        \ Squash
                            (Id
                                (Dom
                                    (ListRel (l1 ⌢ [last])
                                        ▷ {r|c2 dominates R_exist r})))
                          Image ns
                        ∩ {i|R_exist (Nth (l1 ⌢ [last]) i) = c2})⌝);
a(asm_rewrite_tac[extract_def,rel_list_null_thm]);
```

SML

```
(* *** Goal "1.2.1.2" *** *)
a(fc_tac[↑_extract_null_thm]);
a(spec_nth_asm_tac 1⌜(1 .. # (l1 ⌢ [last])
                        \ Squash
                            (Id
                                (Dom
                                    (ListRel (l1 ⌢ [last])
                                        ▷ {r|c2 dominates R_exist r})))
                          Image ns
                        ∩ {i|R_exist (Nth (l1 ⌢ [last]) i) = c2})⌝);
a(asm_rewrite_tac[extract_def,rel_list_null_thm]);
```

SML

```
(* *** Goal "1.2.2" *** *)
a(REPEAT ∀_tac THEN cases_tac⌜c1 dominates R_exist last⌝THEN
        cases_tac⌜c1 dominates R_exist last'⌝ THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2.1" *** *)
a(cases_tac⌜c2 dominates R_exist last⌝
        THEN cases_tac⌜c2 dominates R_exist last'⌝
        THEN asm_rewrite_tac[size_list_rel_⌢_▷_thm]);
(* *** Goal "1.2.2.1.1" *** *)
a(REPEAT ⇒_tac THEN DROP_NTH_ASM_T 10(ante_tac o ∀_elim⌜l2⌝)
        THEN DROP_NTH_ASM_T 9(fn _ => id_tac)
        THEN asm_rewrite_tac[extract_⌢_single_lemma] THEN ⇒_tac);
```

SML

```
a(LEMMA_T⌜# l1 + 1
                    ∈ 1 .. # (l1 ⌢ [last'])
                      \ Squash
                          (Id
                              (Dom
                                  (ListRel (l1 ⌢ [last'])
                                      ▷ {r|c2 dominates R_exist r})))
                        Image ns
                       ∩ {i|R_exist (Nth (l1 ⌢ [last']) i) = c2} ⇔
      # l2 + 1
                    ∈ 1 .. # (l2 ⌢ [last])
                      \ Squash
                          (Id
                              (Dom
                                  (ListRel (l2 ⌢ [last])
                                      ▷ {r|c2 dominates R_exist r})))
                        Image ns
                       ∩ {i|R_exist (Nth (l2 ⌢ [last]) i) = c2}⌝asm_tac);
```

SML

```
(* *** Goal "1.2.2.1.1.1" *** *)
a(POP_ASM_T (fn _ => id_tac) THEN POP_ASM_T (fn _ => id_tac)
      THEN DROP_NTH_ASM_T 2 (strip_asm_tac o rewrite_rule
      [cleanRow_def,get_spec⌜MkRow⌝,row_components,rel_ext_clauses]));
a(POP_ASM_T (fn _ => id_tac));
a(fc_tac[size_squash_id_dom_thm]);
a(asm_rewrite_tac[squash_⌢_thm,image_def,dot_dot_def,≤_plus1_thm,
      length_⌢_one_thm]);
a(REPEAT strip_tac);
(* *** Goal "1.2.2.1.1.1.1" *** *)
a(POP_ASM_T (strip_asm_tac o rewrite_rule[squash_def,list_rel_def,
      enumerate_def,id_def,dot_dot_def]));
a(DROP_NTH_ASM_T 8 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2.1.1.1.2" *** *)
a(spec_nth_asm_tac 3 ⌜x⌝);
a(POP_ASM_T ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2.1.1.1.3" *** *)
a(POP_ASM_T (strip_asm_tac o rewrite_rule[squash_def,list_rel_def,
      enumerate_def,id_def,dot_dot_def]));
a(DROP_NTH_ASM_T 8 ante_tac THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "1.2.2.1.1.1.4" *** *)
a(DROP_NTH_ASM_T 3 ante_tac THEN DROP_NTH_ASM_T 2 ante_tac
        THEN asm_rewrite_tac[nth_length_one_thm]THEN REPEAT strip_tac);
(* *** Goal "1.2.2.1.1.1.5" *** *)
a(POP_ASM_T (strip_asm_tac o rewrite_rule[squash_def,list_rel_def,
        enumerate_def,id_def,dot_dot_def]));
a(DROP_NTH_ASM_T 8 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2.1.1.1.6" *** *)
a(spec_nth_asm_tac 3 ⌜x⌝);
a(POP_ASM_T ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2.1.1.1.7" *** *)
a(DROP_NTH_ASM_T 3 ante_tac THEN DROP_NTH_ASM_T 2 ante_tac
        THEN asm_rewrite_tac[nth_length_one_thm]THEN REPEAT strip_tac);
(* *** Goal "1.2.2.1.1.1.8" *** *)
a(DROP_NTH_ASM_T 3 ante_tac THEN DROP_NTH_ASM_T 2 ante_tac
        THEN asm_rewrite_tac[nth_length_one_thm]THEN REPEAT strip_tac);
```

SML

```
(* *** Goal "1.2.2.1.1.2" *** *)
a(CASES_T⌜# l1 + 1
              ∈ 1 .. # (l1 ⌢ [last'])
                \ Squash
                    (Id
                        (Dom
                            (ListRel (l1 ⌢ [last'])
                                ▷ {r|c2 dominates R_exist r})))
                  Image ns
                ∩ {i|R_exist (Nth (l1 ⌢ [last']) i) = c2}⌝asm_tac);
(* *** Goal "1.2.2.1.1.2.1" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN asm_rewrite_tac[]
        THEN ⇒_T asm_rewrite_thm_tac);
(* *** Goal "1.2.2.1.1.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN asm_rewrite_tac[]
        THEN ⇒_T asm_rewrite_thm_tac);
```

SML

```
(* *** Goal "1.2.2.1.2" *** *)
a(REPEAT ⇒_tac);
a(fc_tac[map_cleanRow_lemma2] THEN asm_fc_tac[]);
a(DROP_NTH_ASM_T 16 ante_tac THEN DROP_NTH_ASM_T 5 rewrite_thm_tac);
(* *** Goal "1.2.2.1.3" *** *)
a(REPEAT ⇒_tac);
a(fc_tac[map_cleanRow_lemma2] THEN asm_fc_tac[]);
a(DROP_NTH_ASM_T 16 ante_tac THEN DROP_NTH_ASM_T 5 rewrite_thm_tac);
(* *** Goal "1.2.2.1.4" *** *)
a(REPEAT ⇒_tac THEN DROP_NTH_ASM_T 10(ante_tac o ∀_elim⌜l_2⌝)
        THEN DROP_NTH_ASM_T 9(fn _ => id_tac)
        THEN asm_rewrite_tac[extract_⌢_single_lemma] THEN ⇒_tac);
```

SML

```
a(LEMMA_T⌜# l_1 + 1
                    ∈ 1 .. # (l_1 ⌢ [last'])
                      \ Squash
                          (Id
                              (Dom
                                  (ListRel (l_1 ⌢ [last'])
                                      ▷ {r|c_2 dominates R_exist r})))
                        Image ns
                      ∩ {i|R_exist (Nth (l_1 ⌢ [last']) i) = c_2} ⇔
      # l_2 + 1
                    ∈ 1 .. # (l_2 ⌢ [last])
                      \ Squash
                          (Id
                              (Dom
                                  (ListRel (l_2 ⌢ [last])
                                      ▷ {r|c_2 dominates R_exist r})))
                        Image ns
                      ∩ {i|R_exist (Nth (l_2 ⌢ [last]) i) = c_2}⌝asm_tac);
```

SML

```
(* *** Goal "1.2.2.1.4.1" *** *)
a(POP_ASM_T (fn _ => id_tac) THEN POP_ASM_T (fn _ => id_tac)
        THEN DROP_NTH_ASM_T 2 (strip_asm_tac o rewrite_rule
        [cleanRow_def,get_spec⌜MkRow⌝,row_components,rel_ext_clauses]));
a(POP_ASM_T (fn _ => id_tac));
a(fc_tac[size_squash_id_dom_thm]);
a(asm_rewrite_tac[squash_⌒_thm,image_def,dot_dot_def,≤_plus1_thm,
        length_⌒_one_thm]);
a(REPEAT strip_tac);
(* *** Goal "1.2.2.1.4.1.1" *** *)
a(POP_ASM_T (strip_asm_tac o rewrite_rule[squash_def,list_rel_def,
        enumerate_def,id_def,dot_dot_def]));
a(DROP_NTH_ASM_T 8 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2.1.4.1.2" *** *)
a(DROP_NTH_ASM_T 3 ante_tac THEN DROP_NTH_ASM_T 2 ante_tac
        THEN asm_rewrite_tac[nth_length_one_thm]THEN REPEAT strip_tac);
(* *** Goal "1.2.2.1.4.1.3" *** *)
a(POP_ASM_T (strip_asm_tac o rewrite_rule[squash_def,list_rel_def,
        enumerate_def,id_def,dot_dot_def]));
a(DROP_NTH_ASM_T 8 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2.1.4.1.4" *** *)
a(DROP_NTH_ASM_T 3 ante_tac THEN DROP_NTH_ASM_T 2 ante_tac
        THEN asm_rewrite_tac[nth_length_one_thm]THEN REPEAT strip_tac);
```

SML

```
(* *** Goal "1.2.2.1.4.2" *** *)
a(CASES_T⌜¬ # l1 + 1
                ∈ 1 .. # (l1 ⌢ [last'])
                  \ Squash
                      (Id
                          (Dom
                              (ListRel (l1 ⌢ [last'])
                                  ▷ {r|c2 dominates R_exist r})))
                    Image ns
                  ∩ {i|R_exist (Nth (l1 ⌢ [last']) i) = c2}⌝asm_tac);
(* *** Goal "1.2.2.1.4.2.1" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN asm_rewrite_tac[]
        THEN ⇒_T asm_rewrite_thm_tac);
(* *** Goal "1.2.2.1.4.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN asm_rewrite_tac[]
        THEN ⇒_T asm_rewrite_thm_tac);
```

SML

```
(* *** Goal "1.2.2.2" *** *)
a(cases_tac⌜c2 dominates R_exist last'⌝
        THEN asm_rewrite_tac[size_list_rel_⌢_▷_thm]);
(* *** Goal "1.2.2.2.1" *** *)
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.2.2.2.2" *** *)
a(REPEAT ⇒_tac THEN DROP_NTH_ASM_T 8(ante_tac o ∀_elim⌜l2 ⌢ [last]⌝)
        THEN DROP_NTH_ASM_T 7(fn _ => id_tac)
        THEN asm_rewrite_tac[size_list_rel_⌢_▷_thm] THEN ⇒_tac);
a(POP_ASM_T (rewrite_thm_tac o eq_sym_rule));
a(asm_rewrite_tac[extract_⌢_single_lemma]);
a(CASES_T⌜# l1 + 1
                        ∈ 1 .. # (l1 ⌢ [last'])
                          \ Squash
                              (Id
                                  (Dom
                                      (ListRel (l1 ⌢ [last'])
                                          ▷ {r|c2 dominates R_exist r})))
                            Image ns
                          ∩ {i|R_exist (Nth (l1 ⌢ [last']) i) = c2}⌝
        asm_rewrite_thm_tac);
```

SML

```
(* *** Goal "1.2.2.3" *** *)
a(cases_tac⌜c2 dominates R_exist last⌝
        THEN asm_rewrite_tac[size_list_rel_⌢_▷_thm]);
(* *** Goal "1.2.2.3.1" *** *)
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.2.2.3.2" *** *)
a(REPEAT ⇒_tac THEN DROP_NTH_ASM_T 8 (fn _ => id_tac)
        THEN DROP_NTH_ASM_T 7(ante_tac o ∀_elim⌜last'⌝)
        THEN asm_rewrite_tac[size_list_rel_⌢_▷_thm] THEN ⇒_tac);
a(POP_ASM_T rewrite_thm_tac);
a(asm_rewrite_tac[extract_⌢_single_lemma]);
a(CASES_T⌜# l2 + 1
                        ∈ 1 .. # (l2 ⌢ [last])
                          \ Squash
                              (Id
                                  (Dom
                                      (ListRel (l2 ⌢ [last])
                                          ▷ {r|c2 dominates R_exist r})))
                                Image ns
                              ∩ {i|R_exist (Nth (l2 ⌢ [last]) i) = c2}⌝
        asm_rewrite_thm_tac);
```

SML

```
(* *** Goal "1.2.2.4" *** *)
a(cases_tac⌜c2 dominates R_exist last⌝
        THEN cases_tac⌜c2 dominates R_exist last'⌝
        THEN asm_rewrite_tac[size_list_rel_⌢_▷_thm]);
```

SML

```
(* *** Goal "1.2.2.4.1" *** *)
a(REPEAT ⇒_tac THEN DROP_NTH_ASM_T 9(ante_tac o ∀_elim⌜l2⌝)
        THEN DROP_NTH_ASM_T 8(fn _ => id_tac)
        THEN asm_rewrite_tac[extract_⌢_single_lemma] THEN ⇒_tac);
a(CASES_T⌜# l1 + 1
                        ∈ 1 .. # (l1 ⌢ [last'])
                          \ Squash
                              (Id
                                  (Dom
                                      (ListRel (l1 ⌢ [last'])
                                          ▷ {r|c2 dominates R_exist r})))
                            Image ns
                           ∩ {i|R_exist (Nth (l1 ⌢ [last']) i) = c2}⌝
        asm_rewrite_thm_tac
        THEN CASES_T⌜# l2 + 1
                        ∈ 1 .. # (l2 ⌢ [last])
                          \ Squash
                              (Id
                                  (Dom
                                      (ListRel (l2 ⌢ [last])
                                          ▷ {r|c2 dominates R_exist r})))
                            Image ns
                           ∩ {i|R_exist (Nth (l2 ⌢ [last]) i) = c2}⌝
        asm_rewrite_thm_tac THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "1.2.2.4.2" *** *)
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.2.2.4.3" *** *)
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.2.2.4.4" *** *)
a(REPEAT ⇒_tac THEN DROP_NTH_ASM_T 9(ante_tac o ∀_elim⌜l2⌝)
        THEN DROP_NTH_ASM_T 8(fn _ => id_tac)
        THEN asm_rewrite_tac[extract_⌢_single_lemma] THEN ⇒_tac);
a(CASES_T⌜# l1 + 1
                        ∈ 1 .. # (l1 ⌢ [last'])
                          \ Squash
                              (Id
                                  (Dom
                                      (ListRel (l1 ⌢ [last'])
                                          ▷ {r|c2 dominates R_exist r})))
                            Image ns
                           ∩ {i|R_exist (Nth (l1 ⌢ [last']) i) = c2}⌝
        asm_rewrite_thm_tac
        THEN CASES_T⌜# l2 + 1
                        ∈ 1 .. # (l2 ⌢ [last])
                          \ Squash
                              (Id
                                  (Dom
                                      (ListRel (l2 ⌢ [last])
                                          ▷ {r|c2 dominates R_exist r})))
                            Image ns
                           ∩ {i|R_exist (Nth (l2 ⌢ [last]) i) = c2}⌝
        asm_rewrite_thm_tac THEN asm_rewrite_tac[]);
val cleanTable_deleteRows_lemma = save_pop_thm"cleanTable_deleteRows_lemma";
```

HOL output

```
cleanTable_deleteRows_lemma =
⊢ ∀ c1 c2 t1 t2 ns
  • cleanTable c1 t1 = cleanTable c1 t2
        ∧ c1 dominates c2
        ∧ c2 dominates TS_class t1
      ⇒ cleanTable
          c1
          (replaceRows
              t1
              (Extract
                  (1 .. # (TS_rows t1)
                      \ revealRow c2 t1 Image ns
                        ∩ {i|R_exist (Nth (TS_rows t1) i) = c2})
                  (TS_rows t1)))
        = cleanTable
          c1
          (replaceRows
              t2
              (Extract
                  (1 .. # (TS_rows t2)
                      \ revealRow c2 t2 Image ns
                        ∩ {i|R_exist (Nth (TS_rows t2) i) = c2})
                  (TS_rows t2)))
```

3.3 Update Rows Lemma

We prove two subsidiary results for *updateField* and *updateRow*.

3.3.1 Update Field Lemma

SML

```
push_goal([],⌜∀ c1 c2 d1 d2 t u •
        c1 dominates c2
        ∧ replaceData c1 d1 = replaceData c1 d2
        ∧ isVal(updateField c2 (TS_class t) (u,d1))
        ∧ isVal(updateField c2 (TS_class t) (u,d2))
        ⇒
        replaceData c1 (destVal(updateField c2 (TS_class t) (u,d1)))
        =
        replaceData c1 (destVal(updateField c2 (TS_class t) (u,d2)))⌝);
a(REPEAT strip_tac);
a(POP_ASM_T ante_tac THEN POP_ASM_T ante_tac THEN rewrite_tac[updateField_def]);
a(cases_tac⌜c2 = TS_class t⌝
        THEN cases_tac⌜isItem u⌝
        THEN cases_tac⌜isClass u⌝
        THEN asm_rewrite_tac[¬isVal_giveError_thm,destVal_def]);
```

SML

```
(* *** Goal "1" *** *)
a(⇒_T (fn _ => id_tac) THEN ⇒_T (fn _ => id_tac));
a(DROP_NTH_ASM_T 4 ante_tac THEN
        asm_rewrite_tac[replaceData_def,get_spec⌜MkData⌝]);
a(cases_tac⌜c1 dominates Dat_class d1⌝ THEN cases_tac⌜c1 dominates Dat_class d2⌝
        THEN asm_rewrite_tac[]);
(* *** Goal "1.1" *** *)
a(⇒_T rewrite_thm_tac);
(* *** Goal "1.2" *** *)
a(rewrite_tac[get_spec⌜MkData⌝,data_components] THEN strip_tac);
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.3" *** *)
a(rewrite_tac[get_spec⌜MkData⌝,data_components] THEN strip_tac);
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "2" *** *)
a(⇒_T (fn _ => id_tac) THEN ⇒_T (fn _ => id_tac));
a(DROP_NTH_ASM_T 4 ante_tac THEN
        asm_rewrite_tac[replaceData_def,get_spec⌜MkData⌝]);
a(cases_tac⌜c1 dominates Dat_class d1⌝ THEN cases_tac⌜c1 dominates Dat_class d2⌝
        THEN asm_rewrite_tac[]);
(* *** Goal "2.1" *** *)
a(⇒_T rewrite_thm_tac);
(* *** Goal "2.2" *** *)
a(rewrite_tac[get_spec⌜MkData⌝,data_components] THEN strip_tac);
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "2.3" *** *)
a(rewrite_tac[get_spec⌜MkData⌝,data_components] THEN strip_tac);
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "3" *** *)
a(cases_tac⌜destClass u dominates Dat_class d1⌝
        THEN cases_tac⌜destClass u dominates Dat_class d2⌝
        THEN asm_rewrite_tac[¬isVal_giveError_thm,destVal_def]);
a(⇒_T (fn _ => id_tac) THEN ⇒_T (fn _ => id_tac));
a(DROP_NTH_ASM_T 6 ante_tac THEN
        asm_rewrite_tac[replaceData_def,get_spec⌜MkData⌝]);
a(cases_tac⌜c1 dominates destClass u⌝THEN asm_rewrite_tac[]);
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
a(asm_rewrite_tac[]);
a(⇒_T rewrite_thm_tac);
```

SML

```
(* *** Goal "4" *** *)
a(cases_tac⌜Dat_class d1 dominates c2⌝
        THEN cases_tac⌜Dat_class d2 dominates c2⌝
        THEN asm_rewrite_tac[¬isVal_giveError_thm,destVal_def]);
a(⇒_T (fn _ => id_tac) THEN ⇒_T (fn _ => id_tac));
a(DROP_NTH_ASM_T 6 ante_tac THEN
        asm_rewrite_tac[replaceData_def,get_spec⌜MkData⌝]);
a(cases_tac⌜c1 dominates Dat_class d1⌝ THEN cases_tac⌜c1 dominates Dat_class d2⌝
        THEN asm_rewrite_tac[]);
(* *** Goal "4.1" *** *)
a(⇒_T rewrite_thm_tac);
(* *** Goal "4.2" *** *)
a(rewrite_tac[get_spec⌜MkData⌝,data_components]THEN strip_tac);
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "4.3" *** *)
a(rewrite_tac[get_spec⌜MkData⌝,data_components]THEN strip_tac);
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "5" *** *)
a(cases_tac⌜Dat_class d1 dominates c2⌝
        THEN cases_tac⌜Dat_class d2 dominates c2⌝
        THEN asm_rewrite_tac[¬isVal_giveError_thm,destVal_def]);
a(⇒_T (fn _ => id_tac) THEN ⇒_T (fn _ => id_tac));
a(DROP_NTH_ASM_T 6 ante_tac THEN
        asm_rewrite_tac[replaceData_def,get_spec⌜MkData⌝]);
a(cases_tac⌜c1 dominates Dat_class d1⌝ THEN cases_tac⌜c1 dominates Dat_class d2⌝
        THEN asm_rewrite_tac[]);
(* *** Goal "5.1" *** *)
a(⇒_T rewrite_thm_tac);
(* *** Goal "5.2" *** *)
a(rewrite_tac[get_spec⌜MkData⌝,data_components]THEN strip_tac);
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "5.3" *** *)
a(rewrite_tac[get_spec⌜MkData⌝,data_components]THEN strip_tac);
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
val replaceData_updateField_lemma = save_pop_thm"replaceData_updateField_lemma";
```

HOL output

```
replaceData_updateField_lemma =
⊢ ∀ c1 c2 d1 d2 t u
  • c1 dominates c2
       ∧ replaceData c1 d1 = replaceData c1 d2
       ∧ isVal (updateField c2 (TS_class t) (u, d1))
       ∧ isVal (updateField c2 (TS_class t) (u, d2))
     ⇒ replaceData c1 (destVal (updateField c2 (TS_class t) (u, d1)))
       = replaceData c1 (destVal (updateField c2 (TS_class t) (u, d2)))
```

3.3.2 Update Row Lemma

SML

```
push_goal([],⌜∀ c1 c2 r1 r2 t u
       • c1 dominates c2
       ∧ Dom u ⊆ {n|∃ c'• c' ∈ visibleCols c2 t ∧ CS_posn c' = n}
       ∧ cleanRow c1 (Snd (cleanColCons c1 t)) r1
           = cleanRow c1 (Snd (cleanColCons c1 t)) r2
       ∧ isVal (updateRow c2 (TS_class t) (u, r1))
       ∧ isVal (updateRow c2 (TS_class t) (u, r2))
       ⇒ cleanRow c1(Snd (cleanColCons c1 t))(destVal(updateRow c2 (TS_class t)(u, r1)))
       = cleanRow c1(Snd (cleanColCons c1 t))(destVal(updateRow c2 (TS_class t)(u, r2)))⌝);
a(REPEAT strip_tac);
a(POP_ASM_T ante_tac THEN POP_ASM_T ante_tac THEN rewrite_tac[updateRow_def]);
```

SML

```
a(cases_tac⌜¬ u ∈ Functional⌝ THEN
       cases_tac⌜((RelCombine u (R_data r1)
                                 ⨾ Graph (updateField c2 (TS_class t)))
                               ▷ {x|isError x})
                              ⨾ Graph destError = {}⌝
       THEN cases_tac⌜((RelCombine u (R_data r2)
                                 ⨾ Graph (updateField c2 (TS_class t)))
                               ▷ {x|isError x})
                              ⨾ Graph destError = {}⌝
       THEN asm_rewrite_tac[¬isVal_giveError_thm,giveVal_eq_thm]);
a(⇒_T (fn _ => id_tac) THEN ⇒_T (fn _ => id_tac) THEN rewrite_tac[destVal_def]);
a(DROP_NTH_ASM_T 3 (fn _ => id_tac));
a(DROP_NTH_ASM_T 3 ante_tac THEN rewrite_tac[cleanRow_def,filterRow_def,
       get_spec⌜MkRow⌝,row_components,rel_ext_clauses,rel_combine_def,⊕_thm]
       THEN REPEAT ⇒_tac THEN strip_tac THEN_TRY asm_rewrite_tac[]);
a(REPEAT strip_tac);
```

SML

```
(* *** Goal "1" *** *)
(* no update *)
a(lemma_tac⌜¬ ∃ up • (x,up) ∈ u⌝);
(* *** Goal "1.1" *** *)
a contr_tac;
a(spec_nth_asm_tac 4 ⌜destVal(updateField c2 (TS_class t) (up,z))⌝);
a(spec_nth_asm_tac 1 ⌜updateField c2 (TS_class t) (up,z)⌝);
a(spec_nth_asm_tac 1 ⌜(up,z)⌝);
(* *** Goal "1.1.1" *** *)
a(POP_ASM_T (strip_asm_tac o rewrite_rule[]));
(* *** Goal "1.1.2" *** *)
a(POP_ASM_T (strip_asm_tac o rewrite_rule[]));
```

SML

```
(* *** Goal "1.2" *** *)
a(DROP_NTH_ASM_T 7(asm_tac o list_∀_elim[⌜x⌝,⌜replaceData c1 z⌝]));
a(LEMMA_T⌜(∃ z'
            • ((∃ c• c ∈ Snd (cleanColCons c1 t) ∧ CS_posn c = x)
                ∧ (x, z') ∈ R_data r1)
              ∧ replaceData c1 z = replaceData c1 z')⌝asm_tac);
(* *** Goal "1.2.1" *** *)
a(∃_tac⌜z⌝THEN asm_rewrite_tac[]);
a(∃_tac⌜c⌝THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN asm_rewrite_tac[] THEN ⇒_tac);
a(∃_tac⌜z'⌝THEN asm_rewrite_tac[]);
a(LEMMA_T⌜∃ c• c ∈ Snd (cleanColCons c1 t) ∧ CS_posn c = x⌝rewrite_thm_tac);
(* *** Goal "1.2.2.1" *** *)
a(∃_tac⌜c⌝THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2.2" *** *)
a(REPEAT strip_tac);
a(spec_nth_asm_tac 9 ⌜Fst z'''⌝);
```

SML

```
(* *** Goal "2" *** *)
(* update *)
a(DROP_NTH_ASM_T 8(asm_tac o list_∀_elim[⌜x⌝,⌜replaceData c1 (Snd z'')⌝]));
a(LEMMA_T⌜(∃ z
          • ((∃ c• c ∈ Snd (cleanColCons c1 t) ∧ CS_posn c = x)
               ∧ (x, z) ∈ R_data r1)
             ∧ replaceData c1 (Snd z'') = replaceData c1 z)⌝asm_tac);
(* *** Goal "2.1" *** *)
a(∃_tac⌜Snd z''⌝THEN asm_rewrite_tac[]);
a(∃_tac⌜c⌝THEN asm_rewrite_tac[]);
(* *** Goal "2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN asm_rewrite_tac[] THEN ⇒_tac);
a(∃_tac⌜destVal(updateField c2 (TS_class t)(Fst z'',z'''))⌝);
a(LEMMA_T⌜∃ c• c ∈ Snd (cleanColCons c1 t) ∧ CS_posn c = x⌝rewrite_thm_tac);
(* *** Goal "2.2.1" *** *)
a(∃_tac⌜c⌝THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "2.2.2" *** *)
a(LEMMA_T⌜(¬ (∃ y z
                • (∃ z'
                    • ((x, Fst z') ∈ u ∧ (x, Snd z') ∈ R_data r2)
                         ∧ z = updateField c2 (TS_class t) z')
                  ∧ y = destVal z)
               ∧ (x, destVal (updateField c2 (TS_class t) (Fst z'', z''')))
                 ∈ R_data r2
             ∨ (∃ z'
             • (∃ z
                • ((x, Fst z) ∈ u ∧ (x, Snd z) ∈ R_data r2)
                     ∧ z' = updateField c2 (TS_class t) z)
                ∧ destVal (updateField c2 (TS_class t) (Fst z'', z'''))
                  = destVal z'))⌝rewrite_thm_tac);
```

SML

```
(* *** Goal "2.2.2.1" *** *)
a ∨_right_tac;
a(∃_tac⌜updateField c2 (TS_class t) (Fst z'', z''')⌝THEN rewrite_tac[]);
a(∃_tac⌜(Fst z'', z''')⌝THEN asm_rewrite_tac[]);
(* *** Goal "2.2.2.2" *** *)
a(lemma_tac⌜isVal (updateField c2 (TS_class t) (Fst z'', z''')) ∧
        isVal (updateField c2 (TS_class t) z'')⌝);
(* *** Goal "2.2.2.2.1" *** *)
a strip_tac;
(* *** Goal "2.2.2.2.1.1" *** *)
a(strip_asm_tac(∀_elim⌜(updateField c2 (TS_class t) (Fst z'', z'''))⌝val_or_error_type));
a(DROP_NTH_ASM_T 16 (strip_asm_tac o rewrite_rule[rel_ext_clauses,rel_combine_def]));
a(list_spec_nth_asm_tac 1 [⌜x⌝,⌜destError(updateField c2 (TS_class t) (Fst z'', z'''))⌝]);
a(spec_nth_asm_tac 1 ⌜(updateField c2 (TS_class t) (Fst z'', z'''))⌝);
a(POP_ASM_T(strip_asm_tac o rewrite_rule[] o ∀_elim ⌜(Fst z'', z''')⌝));
```

SML

```
(* *** Goal "2.2.2.2.1.2" *** *)
a(strip_asm_tac(∀_elim⌜(updateField c2 (TS_class t) z'')⌝val_or_error_type));
a(DROP_NTH_ASM_T 17 (strip_asm_tac o rewrite_rule[rel_ext_clauses,rel_combine_def]));
a(list_spec_nth_asm_tac 1 [⌜x⌝,⌜destError(updateField c2 (TS_class t) z'')⌝]);
a(spec_nth_asm_tac 1 ⌜(updateField c2 (TS_class t) z'')⌝);
a(POP_ASM_T(strip_asm_tac o rewrite_rule[] o ∀_elim ⌜z''⌝));
(* *** Goal "2.2.2.2.2" *** *)
a(asm_rewrite_tac[]);
a(POP_ASM_T ante_tac THEN pure_once_rewrite_tac[prove_rule[]⌜z'' = (Fst z'',Snd z'')⌝]
        THEN ⇒_tac);
a(fc_tac[replaceData_updateField_lemma]
        THEN asm_fc_tac[] THEN asm_fc_tac[] THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "3" *** *)
(* no update *)
a(lemma_tac⌜¬ ∃ up • (x,up) ∈ u⌝);
(* *** Goal "3.1" *** *)
a contr_tac;
a(spec_nth_asm_tac 4 ⌜destVal(updateField c2 (TS_class t) (up,z))⌝);
a(spec_nth_asm_tac 1 ⌜updateField c2 (TS_class t) (up,z)⌝);
a(spec_nth_asm_tac 1 ⌜(up,z)⌝);
(* *** Goal "3.1.1" *** *)
a(POP_ASM_T (strip_asm_tac o rewrite_rule[]));
(* *** Goal "3.1.2" *** *)
a(POP_ASM_T (strip_asm_tac o rewrite_rule[]));
```

SML

```
(* *** Goal "3.2" *** *)
a(DROP_NTH_ASM_T 7(asm_tac o list_∀_elim[⌜x⌝,⌜replaceData c1 z⌝]));
a(LEMMA_T⌜(∃ z'
            • ((∃ c• c ∈ Snd (cleanColCons c1 t) ∧ CS_posn c = x)
                ∧ (x, z') ∈ R_data r2)
              ∧ replaceData c1 z = replaceData c1 z')⌝asm_tac);
(* *** Goal "3.2.1" *** *)
a(∃_tac⌜z⌝THEN asm_rewrite_tac[]);
a(∃_tac⌜c⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN asm_rewrite_tac[] THEN ⇒_tac);
a(∃_tac⌜z'⌝THEN asm_rewrite_tac[]);
a(LEMMA_T⌜∃ c• c ∈ Snd (cleanColCons c1 t) ∧ CS_posn c = x⌝rewrite_thm_tac);
(* *** Goal "3.2.2.1" *** *)
a(∃_tac⌜c⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.2.2" *** *)
a(REPEAT strip_tac);
a(spec_nth_asm_tac 9 ⌜Fst z'''⌝);
```

SML

```
(* *** Goal "4" *** *)
(* update *)
a(DROP_NTH_ASM_T 8(asm_tac o list_∀_elim[⌜x⌝,⌜replaceData c1 (Snd z'')⌝]));
a(LEMMA_T⌜(∃ z
          • ((∃ c• c ∈ Snd (cleanColCons c1 t) ∧ CS_posn c = x)
               ∧ (x, z) ∈ R_data r2)
             ∧ replaceData c1 (Snd z'') = replaceData c1 z)⌝asm_tac);
(* *** Goal "4.1" *** *)
a(∃_tac⌜Snd z''⌝THEN asm_rewrite_tac[]);
a(∃_tac⌜c⌝THEN asm_rewrite_tac[]);
(* *** Goal "4.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN asm_rewrite_tac[] THEN ⇒_tac);
a(∃_tac⌜destVal(updateField c2 (TS_class t)(Fst z'',z'''))⌝);
a(LEMMA_T⌜∃ c• c ∈ Snd (cleanColCons c1 t) ∧ CS_posn c = x⌝rewrite_thm_tac);
(* *** Goal "4.2.1" *** *)
a(∃_tac⌜c⌝THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "4.2.2" *** *)
a(LEMMA_T⌜(¬ (∃ y z
                • (∃ z'
                    • ((x, Fst z') ∈ u ∧ (x, Snd z') ∈ R_data r1)
                         ∧ z = updateField c2 (TS_class t) z')
                    ∧ y = destVal z)
               ∧ (x, destVal (updateField c2 (TS_class t) (Fst z'', z''')))
                 ∈ R_data r1
             ∨ (∃ z'
             • (∃ z
                • ((x, Fst z) ∈ u ∧ (x, Snd z) ∈ R_data r1)
                     ∧ z' = updateField c2 (TS_class t) z)
                ∧ destVal (updateField c2 (TS_class t) (Fst z'', z'''))
                  = destVal z'))⌝rewrite_thm_tac);
```

SML

```
(* *** Goal "4.2.2.1" *** *)
a ∨_right_tac;
a(∃_tac⌜updateField c2 (TS_class t) (Fst z'', z''')⌝THEN rewrite_tac[]);
a(∃_tac⌜(Fst z'', z''')⌝THEN asm_rewrite_tac[]);
(* *** Goal "4.2.2.2" *** *)
a(lemma_tac⌜isVal (updateField c2 (TS_class t) (Fst z'', z''')) ∧
        isVal (updateField c2 (TS_class t) z'')⌝);
(* *** Goal "4.2.2.2.1" *** *)
a strip_tac;
(* *** Goal "4.2.2.2.1.1" *** *)
a(strip_asm_tac(∀_elim⌜(updateField c2 (TS_class t) (Fst z'', z'''))⌝val_or_error_type));
a(DROP_NTH_ASM_T 17 (strip_asm_tac o rewrite_rule[rel_ext_clauses,rel_combine_def]));
a(list_spec_nth_asm_tac 1 [⌜x⌝,⌜destError(updateField c2 (TS_class t) (Fst z'', z'''))⌝]);
a(spec_nth_asm_tac 1 ⌜(updateField c2 (TS_class t) (Fst z'', z'''))⌝);
a(POP_ASM_T(strip_asm_tac o rewrite_rule[] o ∀_elim ⌜(Fst z'', z''')⌝));
```

SML

```
(* *** Goal "4.2.2.2.1.2" *** *)
a(strip_asm_tac(∀_elim⌜(updateField c2 (TS_class t) z'')⌝val_or_error_type));
a(DROP_NTH_ASM_T 16 (strip_asm_tac o rewrite_rule[rel_ext_clauses,rel_combine_def]));
a(list_spec_nth_asm_tac 1 [⌜x⌝,⌜destError(updateField c2 (TS_class t) z'')⌝]);
a(spec_nth_asm_tac 1 ⌜(updateField c2 (TS_class t) z'')⌝);
a(POP_ASM_T(strip_asm_tac o rewrite_rule[] o ∀_elim ⌜z''⌝));
(* *** Goal "4.2.2.2.2" *** *)
a(asm_rewrite_tac[]);
a(POP_ASM_T ante_tac THEN pure_once_rewrite_tac[prove_rule[]⌜z'' = (Fst z'',Snd z'')⌝]
        THEN ⇒_tac);
a(fc_tac[replaceData_updateField_lemma]
        THEN asm_fc_tac[] THEN asm_fc_tac[] THEN asm_rewrite_tac[]);
val cleanRow_updateRow_lemma1 = save_pop_thm"cleanRow_updateRow_lemma1";
```

HOL output

```
cleanRow_updateRow_lemma1 =
⊢ ∀ c1 c2 r1 r2 t u
  • c1 dominates c2
       ∧ Dom u ⊆ {n|∃ c'• c' ∈ visibleCols c2 t ∧ CS_posn c' = n}
       ∧ cleanRow c1 (Snd (cleanColCons c1 t)) r1
         = cleanRow c1 (Snd (cleanColCons c1 t)) r2
       ∧ isVal (updateRow c2 (TS_class t) (u, r1))
       ∧ isVal (updateRow c2 (TS_class t) (u, r2))
     ⇒ cleanRow
         c1
         (Snd (cleanColCons c1 t))
         (destVal (updateRow c2 (TS_class t) (u, r1)))
       = cleanRow
         c1
         (Snd (cleanColCons c1 t))
         (destVal (updateRow c2 (TS_class t) (u, r2)))
```

3.3.3 Update Rows Lemma

We first prove a theorem which essentially states that if the row existence class of two rows is the same and the same fields of the two rows are visible, then an update either exists for both rows or for neither.

SML

```
push_goal([],⌜∀ l1 l2 x1 x2 c us up •
       R_exist x1 = R_exist x2 ∧
       #(ListRel l1 ▷ {r|c dominates R_exist r})
           = #(ListRel l2 ▷ {r|c dominates R_exist r})
       ⇒
       (((#(Squash (Id (Dom (ListRel (l1 ⌢ [x1])
                                     ▷ {r|c dominates R_exist r}))))),up) ∈ us)
       ⇔((#(Squash (Id (Dom (ListRel (l2 ⌢ [x2])
                                     ▷ {r|c dominates R_exist r}))))),up) ∈ us))⌝);
```

SML

```
a(REPEAT ∀_tac THEN strip_tac);
a(cases_tac⌜c dominates R_exist x2⌝);
(* *** Goal "1" *** *)
a(lemma_tac⌜c dominates R_exist x1⌝ THEN_LIST[asm_rewrite_tac[],id_tac]);
a(fc_tac[size_squash_id_dom_thm]);
a(LEMMA_T⌜#(Squash (Id (Dom (ListRel (l1 ⌢ [x1])
                                ▷ {r|c dominates R_exist r}))))
        = #(Squash (Id (Dom (ListRel (l2 ⌢ [x2])
                                ▷ {r|c dominates R_exist r}))))⌝rewrite_thm_tac);
a(strip_asm_tac(list_∀_elim[⌜l1⌝,⌜x1⌝,⌜{r|c dominates R_exist r}⌝]size_squash_plus1_thm));
a(strip_asm_tac(list_∀_elim[⌜l2⌝,⌜x2⌝,⌜{r|c dominates R_exist r}⌝]size_squash_plus1_thm));
a(asm_rewrite_tac[]);
(* *** Goal "2" *** *)
a(lemma_tac⌜¬ c dominates R_exist x1⌝ THEN_LIST[asm_rewrite_tac[],id_tac]);
a(LEMMA_T⌜#(Squash (Id (Dom (ListRel (l1 ⌢ [x1])
                                ▷ {r|c dominates R_exist r}))))
        = #(Squash (Id (Dom (ListRel (l2 ⌢ [x2])
                                ▷ {r|c dominates R_exist r}))))⌝rewrite_thm_tac);
a(asm_rewrite_tac[list_rel_⌢_▷_thm]);
a(fc_tac[size_squash_id_dom_thm]);
val inUpdates_lemma = save_pop_thm"inUpdates_lemma";
```

HOL output

```
|inUpdates_lemma =
|⊢ ∀ l1 l2 x1 x2 c us up
|  • R_exist x1 = R_exist x2
|       ∧ # (ListRel l1 ▷ {r|c dominates R_exist r})
|         = # (ListRel l2 ▷ {r|c dominates R_exist r})
|     ⇒ ((#
|              (Squash
|                 (Id
|                    (Dom
|                       (ListRel (l1 ⌢ [x1])
|                          ▷ {r|c dominates R_exist r}))))), up)
|         ∈ us
|       ⇔ (#
|              (Squash
|                 (Id
|                    (Dom
|                       (ListRel (l2 ⌢ [x2])
|                          ▷ {r|c dominates R_exist r}))))), up)
|         ∈ us)
```

Now the main theorem of this section.

SML

```
push_goal([],⌜∀ c1 c2 t1 t2 us •
        us ∈ Functional
        ∧
        Dom (⋃ (Ran us)) ⊆ {n|∃ c • c ∈ Snd (cleanColCons c2 t2) ∧ CS_posn c = n}
        ∧
        ((RelCombine
                  ((revealRow c2 t1)~ ⨾ us)
                  (ListRel (TS_rows t1))
                ⨾ Graph (updateRow c2 (TS_class t2)))
              ▷ {x|isError x})
            ⨾ Graph destError
          = {}
        ∧ ((RelCombine
                  ((revealRow c2 t2)~ ⨾ us)
                  (ListRel (TS_rows t2))
                ⨾ Graph (updateRow c2 (TS_class t2)))
              ▷ {x|isError x})
            ⨾ Graph destError
          = {}
        ∧ cleanTable c1 t1 = cleanTable c1 t2
        ∧ c1 dominates c2 ∧ c2 dominates TS_class t1
        ⇒
        cleanTable
              c1
              (replaceRows
                  t1
                  (RelList
                      (ListRel (TS_rows t1)
                          ⊕ (RelCombine
                              ((revealRow c2 t1)~ ⨾ us)
                              (ListRel (TS_rows t1))
                            ⨾ Graph (updateRow c2 (TS_class t1)))
                          ⨾ Graph destVal)))
        =
        cleanTable
              c1
              (replaceRows
                  t2
                  (RelList
                      (ListRel (TS_rows t2)
                          ⊕ (RelCombine
                              ((revealRow c2 t2)~ ⨾ us)
                              (ListRel (TS_rows t2))
                            ⨾ Graph (updateRow c2 (TS_class t2)))
                          ⨾ Graph destVal)))⌝);
```

SML

```
a(REPEAT strip_tac);
a(lemma_tac⌜cleanTable c2 t1 = cleanTable c2 t2⌝
        THEN_LIST[fc_tac[cleanTable_lemma]
        THEN asm_fc_tac[],id_tac]);
a(DROP_NTH_ASM_T 4 ante_tac THEN
        rewrite_tac[cleanTable_def,replaceRows_def,get_spec⌜MkTableSpec⌝]);
a(lemma_tac⌜c1 dominates TS_class t1⌝THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(cases_tac⌜c1 dominates TS_class t2⌝THEN asm_rewrite_tac[]);
```

SML

```
set_labelled_goal"2";
(* *** Goal "2" *** *)
a(rewrite_tac[get_spec⌜MkTableSpec⌝,tab_components,cleanColCons_def]
        THEN REPEAT strip_tac);
a(DROP_NTH_ASM_T 7 ante_tac THEN DROP_NTH_ASM_T 5 rewrite_thm_tac);
a(asm_rewrite_tac[]);
```

SML

```
(* *** Goal "1" *** *)
a(⇒_T(strip_asm_tac o rewrite_rule[get_spec⌜MkTableSpec⌝,tab_components]));
a(DROP_NTH_ASM_T 8 ante_tac THEN asm_rewrite_tac[cleanTable_def]);
a(GET_NTH_ASM_T 8 ante_tac THEN GET_NTH_ASM_T 5 rewrite_thm_tac
        THEN ⇒_tac
        THEN asm_rewrite_tac[get_spec⌜MkTableSpec⌝,tab_components] THEN strip_tac);
a(POP_ASM_T ante_tac THEN lemma_tac⌜cleanColCons c2 t1 = cleanColCons c2 t2⌝
        THEN_LIST[pure_once_asm_rewrite_tac[prove_rule[pair_clauses]
        ⌜∀ p • p = (Fst p, Snd p)⌝]THEN asm_rewrite_tac[],
        DROP_NTH_ASM_T 3(fn _ => id_tac)
        THEN DROP_NTH_ASM_T 2(fn _ => id_tac)]);
a(asm_rewrite_tac[] THEN ⇒_tac);
a(lemma_tac⌜cleanColCons c1 t1 = cleanColCons c1 t2⌝
        THEN_LIST[pure_once_asm_rewrite_tac[prove_rule[pair_clauses]
        ⌜∀ p • p = (Fst p, Snd p)⌝]THEN asm_rewrite_tac[],
        DROP_NTH_ASM_T 7(fn _ => id_tac)
        THEN DROP_NTH_ASM_T 6(fn _ => id_tac)]);
```

SML

```
a(DROP_NTH_ASM_T 5 ante_tac THEN asm_rewrite_tac[] THEN ⇒_tac);
a(fc_tac[cleanRows_size_lemma]);
a(LIST_DROP_NTH_ASM_T [1,2,3,4](MAP_EVERY ante_tac));
a(LIST_DROP_NTH_ASM_T [1,2](MAP_EVERY (fn _ => id_tac)));
a(DROP_NTH_ASM_T 9 ante_tac THEN DROP_NTH_ASM_T 8 ante_tac THEN
        rewrite_tac[cleanColCons_def,cleanRows_def,revealRow_def,get_spec⌜MkTableSpec⌝]);
a(REPEAT strip_tac);
a(LIST_DROP_NTH_ASM_T [1,2,3,6,7](MAP_EVERY ante_tac));
a(lemma_tac⌜∃ l1 • TS_rows t1 = l1⌝THEN_LIST
        [prove_∃_tac,POP_ASM_T rewrite_thm_tac]);
a(lemma_tac⌜∃ l2 • TS_rows t2 = l2⌝THEN_LIST
        [prove_∃_tac,POP_ASM_T rewrite_thm_tac]);
a(intro_∀_tac(⌜l2⌝,⌜l2⌝));
a(REV_LIST_INDUCTION_T⌜l1⌝asm_tac);
```

SML

```
(* *** Goal "1.1" *** *)
a(rewrite_tac[map_def,all_∀_intro(eq_sym_rule(all_∀_elim list_rel_list_thm)),
        ⊕_null_thm,rel_list_null_thm]THEN REPEAT strip_tac);
a(LEMMA_T⌜ListRel l2 ▷ {r|c2 dominates R_exist r} = {}⌝rewrite_thm_tac);
(* *** Goal "1.1.1" *** *)
a(DROP_NTH_ASM_T 3 ante_tac THEN rewrite_tac[rel_ext_clauses,▷_def]);
a(REPEAT strip_tac);
a(list_spec_nth_asm_tac 2 [⌜x⌝,⌜y⌝]);
a(swap_nth_asm_concl_tac 1);
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.1.2" *** *)
a(asm_rewrite_tac[⊕_null_thm2,rel_list_def]);
```

SML

```
(* *** Goal "1.2" *** *)
a(REPEAT ∀_tac);
a(intro_∀_tac(⌜last⌝,⌜last⌝) THEN REV_LIST_INDUCTION_T⌜l2⌝asm_tac);
(* *** Goal "1.2.1" *** *)
a(REPEAT ∀_tac THEN cases_tac⌜c1 dominates R_exist last⌝THEN asm_rewrite_tac
        [map_def,⊕_null_thm,rel_list_null_thm,
        all_∀_intro(eq_sym_rule(all_∀_elim list_rel_list_thm))]THEN REPEAT strip_tac);
a(LEMMA_T⌜ListRel (l1 ⌢ [last]) ▷ {r|c2 dominates R_exist r} = {}⌝rewrite_thm_tac);
(* *** Goal "1.2.1.1" *** *)
a(lemma_tac⌜¬ c2 dominates R_exist last⌝);
(* *** Goal "1.2.1.1.1" *** *)
a(swap_nth_asm_concl_tac 4);
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.2.1.1.2" *** *)
a(asm_rewrite_tac[list_rel_⌢_▷_thm]);
a(DROP_NTH_ASM_T 4 ante_tac THEN rewrite_tac[rel_ext_clauses,▷_def]);
a(REPEAT strip_tac);
a(list_spec_nth_asm_tac 2 [⌜x⌝,⌜y⌝]);
a(swap_nth_asm_concl_tac 1);
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.2.1.2" *** *)
a(asm_rewrite_tac[⊕_null_thm2,rel_list_def,list_rel_⌢_▷_thm]);
```

SML

```
(* *** Goal "1.2.2" *** *)
a(REPEAT ∀_tac THEN cases_tac⌜c1 dominates R_exist last⌝THEN
        cases_tac⌜c1 dominates R_exist last'⌝ THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2.1" *** *)
a(cases_tac⌜c2 dominates R_exist last⌝
        THEN cases_tac⌜c2 dominates R_exist last'⌝
        THEN asm_rewrite_tac[size_list_rel_⌢_▷_thm]);
```

SML

```
(* *** Goal "1.2.2.1.1" *** *)
a(REPEAT ⇒_tac THEN fc_tac[rel_combine_null_lemma]);
a(DROP_NTH_ASM_T 14(ante_tac o ∀_elim⌜l₂⌝)
        THEN DROP_NTH_ASM_T 13(fn _ => id_tac)
        THEN asm_rewrite_tac[] THEN ⇒_tac);
a(GET_NTH_ASM_T 6 (strip_asm_tac o rewrite_rule
        [cleanRow_def,get_spec⌜MkRow⌝,row_components,rel_ext_clauses]));
a(POP_ASM_T (fn _ => id_tac));
a(ante_tac(list_∀_elim[⌜l₁⌝,⌜l₂⌝,⌜last'⌝,⌜last⌝,⌜c₂⌝,⌜us⌝]inUpdates_lemma)
        THEN asm_rewrite_tac[] THEN ⇒_tac);
a(cases_tac⌜∃ up • (#(Squash (Id (Dom (ListRel (l₁ ⌢ [last'])
                                        ▷ {r|c₂ dominates R_exist r}))))),up) ∈ us⌝);
(* *** Goal "1.2.2.1.1.1" *** *)
a(strip_asm_tac(list_∀_elim[
        ⌜#(Squash(Id(Dom(ListRel (l₁ ⌢ [last']) ▷ {r|c₂ dominates R_exist r}))))⌝,
        ⌜up⌝,⌜us⌝,⌜c₂⌝,⌜t₂⌝]dom_⋃_ran_lemma));
a(DROP_NTH_ASM_T 3 (ante_tac o ∀_elim⌜up⌝) THEN asm_rewrite_tac[]THEN ⇒_tac);
a(strip_asm_tac(list_∀_elim[⌜c₂⌝,⌜last'⌝,⌜l₁⌝,⌜up⌝,⌜us⌝,⌜t₂⌝]conjunct1_lemma2)
        THEN POP_ASM_T rewrite_thm_tac);
a(strip_asm_tac(list_∀_elim[⌜c₂⌝,⌜last⌝,⌜l₂⌝,⌜up⌝,⌜us⌝,⌜t₂⌝]conjunct1_lemma2)
        THEN POP_ASM_T rewrite_thm_tac);
a(lemma_tac⌜isVal (updateRow c₂ (TS_class t₂) (up, last'))
        ∧ isVal (updateRow c₂ (TS_class t₂) (up, last))⌝);
```

SML

```
(* *** Goal "1.2.2.1.1.1.1" *** *)
a strip_tac;
(* *** Goal "1.2.2.1.1.1.1.1" *** *)
a(strip_asm_tac(∀_elim⌜(updateRow c2 (TS_class t2) (up, last'))⌝val_or_error_type));
a(DROP_NTH_ASM_T 14 (strip_asm_tac o rewrite_rule[rel_ext_clauses,rel_combine_def]));
a(list_spec_nth_asm_tac 1 [⌜# l1 + 1⌝,⌜destError(updateRow c2(TS_class t2)(up,last'))⌝]);
a(spec_nth_asm_tac 1 ⌜(updateRow c2 (TS_class t2) (up, last'))⌝);
a(POP_ASM_T(strip_asm_tac o rewrite_rule[] o ∀_elim ⌜(up, last')⌝));
(* *** Goal "1.2.2.1.1.1.1.1.1" *** *)
a(spec_nth_asm_tac 1 ⌜#(Squash (Id (Dom (ListRel (l1 ⌢ [last'])
                                        ▷ {r|c2 dominates R_exist r}))))⌝);
a(POP_ASM_T ante_tac THEN asm_rewrite_tac[inv_rel_def]);
a(strip_asm_tac(list_∀_elim[⌜l1⌝,⌜last'⌝,⌜{r|c2 dominates R_exist r}⌝]size_⌢_one_thm));
(* *** Goal "1.2.2.1.1.1.1.2" *** *)
a(POP_ASM_T(strip_asm_tac o rewrite_rule[list_rel_def,dot_dot_def,
        length_⌢_one_thm,nth_length_one_thm]));
```

SML

```
(* *** Goal "1.2.2.1.1.1.2" *** *)
a(strip_asm_tac(∀_elim⌜(updateRow c2 (TS_class t2) (up, last))⌝val_or_error_type));
a(DROP_NTH_ASM_T 15 (strip_asm_tac o rewrite_rule[rel_ext_clauses,rel_combine_def]));
a(list_spec_nth_asm_tac 1 [⌜# l2 + 1⌝,⌜destError(updateRow c2(TS_class t2)(up, last))⌝]);
a(spec_nth_asm_tac 1 ⌜(updateRow c2 (TS_class t2) (up, last))⌝);
a(POP_ASM_T(strip_asm_tac o rewrite_rule[] o ∀_elim ⌜(up, last)⌝));
(* *** Goal "1.2.2.1.1.1.2.1" *** *)
a(spec_nth_asm_tac 1 ⌜#(Squash (Id (Dom (ListRel (l2 ⌢ [last])
                                        ▷ {r|c2 dominates R_exist r}))))⌝);
a(POP_ASM_T ante_tac THEN asm_rewrite_tac[inv_rel_def]);
a(strip_asm_tac(list_∀_elim[⌜l2⌝,⌜last⌝,⌜{r|c2 dominates R_exist r}⌝]size_⌢_one_thm));
(* *** Goal "1.2.2.1.1.1.2.2" *** *)
a(POP_ASM_T(strip_asm_tac o rewrite_rule[list_rel_def,dot_dot_def,
        length_⌢_one_thm,nth_length_one_thm]));
```

SML

```
(* *** Goal "1.2.2.1.1.1.2" *** *)
a(LEMMA_T⌜c1 dominates R_exist
                    (destVal (updateRow c2 (TS_class t2) (up, last')))
      ∧ c1 dominates R_exist
                    (destVal (updateRow c2 (TS_class t2) (up, last)))⌝asm_rewrite_thm_tac);
(* *** Goal "1.2.2.1.1.1.2.1" *** *)
a(POP_ASM_T ante_tac THEN POP_ASM_T ante_tac THEN rewrite_tac[updateRow_def]);
a(cases_tac⌜¬ up ∈ Functional⌝ THEN
      cases_tac⌜((RelCombine up (R_data last')
                          ⨾ Graph (updateField c2 (TS_class t2)))
                        ▷ {x|isError x})
                      ⨾ Graph destError
                     = {}
      ∧ ((RelCombine up (R_data last)
                          ⨾ Graph (updateField c2 (TS_class t2)))
                        ▷ {x|isError x})
                      ⨾ Graph destError
                     = {}⌝
      THEN asm_rewrite_tac[¬isVal_giveError_thm,destVal_def,get_spec⌜MkRow⌝,
      row_components]THEN REPEAT strip_tac);
(* *** Goal "1.2.2.1.1.1.2.2" *** *)
a(bc_tac[rewrite_rule[cleanColCons_def]cleanRow_updateRow_lemma1]THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "1.2.2.1.1.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN asm_rewrite_tac[] THEN ⇒_tac);
a(LEMMA_T⌜RelCombine ((Squash
                              (Id
                                  (Dom
                                      (ListRel (l1 ⌢ [last'])
                                          ▷ {r|c2 dominates R_exist r})))~
                          ⨾ us)
                        (ListRel (l1 ⌢ [last']))
      =RelCombine ((Squash
                              (Id
                                  (Dom
                                      (ListRel l1
                                          ▷ {r|c2 dominates R_exist r})))~
                          ⨾ us)
                        (ListRel l1)⌝rewrite_thm_tac);
```

SML

```
(* *** Goal "1.2.2.1.1.2.1" *** *)
a(asm_rewrite_tac[squash_⌢_thm]);
a(LEMMA_T⌜{(# (Squash(Id(Dom (ListRel l1
        ▷ {r|c2 dominates R_exist r}))))) + 1, # l1 + 1)}~
                ⁰/₉ us = {}⌝rewrite_thm_tac);
(* *** Goal "1.2.2.1.1.2.1.1" *** *)
a(rewrite_tac[rel_ext_clauses,inv_rel_def]THEN REPEAT strip_tac);
a(spec_nth_asm_tac 4 ⌜y⌝);
a(DROP_NTH_ASM_T 4(fn _ => id_tac) THEN DROP_NTH_ASM_T 4(fn _ => id_tac));
a(strip_asm_tac(list_∀_elim[⌜l1⌝,⌜last'⌝,⌜{r|c2 dominates R_exist r}⌝]size_squash_plus1_thm));
a(swap_nth_asm_concl_tac 2);
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2.1.1.2.1.2" *** *)
a(rewrite_tac[rel_combine_one_lemma]);
```

SML

```
(* *** Goal "1.2.2.1.1.2.2" *** *)
a(LEMMA_T`RelCombine ((Squash
                          (Id
                              (Dom
                                  (ListRel (l2 ^ [last])
                                      |> {r|c2 dominates R_exist r})))))~
                      o us)
                  (ListRel (l2 ^ [last]))
    =RelCombine ((Squash
                          (Id
                              (Dom
                                  (ListRel l2
                                      |> {r|c2 dominates R_exist r})))))~
                      o us)
                  (ListRel l2)`rewrite_thm_tac);
(* *** Goal "1.2.2.1.1.2.2.1" *** *)
a(asm_rewrite_tac[squash_^_thm]);
a(LEMMA_T`{(# (Squash(Id(Dom (ListRel l2
        |> {r|c2 dominates R_exist r})))) + 1, # l2 + 1)}~
              o us = {}`rewrite_thm_tac);
(* *** Goal "1.2.2.1.1.2.2.1.1" *** *)
a(rewrite_tac[rel_ext_clauses,inv_rel_def]THEN REPEAT strip_tac);
a(spec_nth_asm_tac 3 `y`);
a(DROP_NTH_ASM_T 4(fn _ => id_tac) THEN DROP_NTH_ASM_T 4(fn _ => id_tac));
a(strip_asm_tac(list_∀_elim[`l2`,`last`,`{r|c2 dominates R_exist r}`]size_squash_plus1_thm));
a(swap_nth_asm_concl_tac 2);
a(DROP_NTH_ASM_T 4 ante_tac THEN asm_rewrite_tac[]);
(* *** Goal "1.2.2.1.1.2.2.1.2" *** *)
a(rewrite_tac[rel_combine_one_lemma]);
(* *** Goal "1.2.2.1.1.2.2.2" *** *)
a(strip_asm_tac(list_∀_elim[`l1`,`last'`,`c2`,`us`,`t2`]conjunct1_lemma1));
a(strip_asm_tac(list_∀_elim[`l2`,`last`,`c2`,`us`,`t2`]conjunct1_lemma1));
a(asm_rewrite_tac[]);
```

SML

```
(* *** Goal "1.2.2.1.2" *** *)
a(REPEAT ⇒_tac);
a(strip_asm_tac(list_∀_elim[⌜l1⌝,⌜l2⌝,⌜c1⌝,⌜c2⌝,⌜{col
                  |∃ y
                    • c1 dominates CC_exist y
                        ∧ (CS_consGroup col, y) ∈ TS_cons t2}⌝]
        map_cleanRow_lemma2));
a(DROP_NTH_ASM_T 3 ante_tac THEN POP_ASM_T rewrite_thm_tac);
```

SML

```
(* *** Goal "1.2.2.1.3" *** *)
a(REPEAT ⇒_tac);
a(strip_asm_tac(list_∀_elim[⌜l1⌝,⌜l2⌝,⌜c1⌝,⌜c2⌝,⌜{col
                  |∃ y
                    • c1 dominates CC_exist y
                        ∧ (CS_consGroup col, y) ∈ TS_cons t2}⌝]
        map_cleanRow_lemma2));
a(DROP_NTH_ASM_T 3 ante_tac THEN POP_ASM_T rewrite_thm_tac);
```

SML

```
(* *** Goal "1.2.2.1.4" *** *)
a(REPEAT ⇒_tac THEN fc_tac[rel_combine_null_lemma]);
a(DROP_NTH_ASM_T 14(ante_tac o ∀_elim⌜l2⌝)
        THEN DROP_NTH_ASM_T 13(fn _ => id_tac)
        THEN asm_rewrite_tac[] THEN ⇒_tac);
a(asm_rewrite_tac[list_rel_⌒_▷_thm,rel_combine_one_lemma]);
a(strip_asm_tac(list_∀_elim[⌜l1⌝,⌜last'⌝,⌜c2⌝,⌜us⌝,⌜t2⌝]conjunct1_lemma1));
a(strip_asm_tac(list_∀_elim[⌜l2⌝,⌜last⌝,⌜c2⌝,⌜us⌝,⌜t2⌝]conjunct1_lemma1));
a(asm_rewrite_tac[]);
```

SML

```
(* *** Goal "1.2.2.2" *** *)
a(cases_tac⌜c2 dominates R_exist last'⌝
        THEN asm_rewrite_tac[size_list_rel_⌢_▷_thm]);
(* *** Goal "1.2.2.2.1" *** *)
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.2.2.2.2" *** *)
a(REPEAT ⇒_tac THEN fc_tac[rel_combine_null_lemma]);
a(DROP_NTH_ASM_T 12(ante_tac o ∀_elim⌜l2 ⌢ [last]⌝)
        THEN DROP_NTH_ASM_T 11(fn _ => id_tac)
        THEN asm_rewrite_tac[size_list_rel_⌢_▷_thm] THEN ⇒_tac);
a(POP_ASM_T (rewrite_thm_tac o eq_sym_rule));
a(asm_rewrite_tac[list_rel_⌢_▷_thm,rel_combine_one_lemma]);
a(strip_asm_tac(list_∀_elim[⌜l1⌝,⌜last'⌝,⌜c2⌝,⌜us⌝,⌜t2⌝]conjunct1_lemma1));
a(asm_rewrite_tac[]);
```

SML

```
(* *** Goal "1.2.2.3" *** *)
a(cases_tac⌜c2 dominates R_exist last⌝
        THEN asm_rewrite_tac[size_list_rel_⌢_▷_thm]);
(* *** Goal "1.2.2.3.1" *** *)
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.2.2.3.2" *** *)
a(REPEAT ⇒_tac THEN fc_tac[rel_combine_null_lemma]);
a(DROP_NTH_ASM_T 12(fn _ => id_tac)
        THEN DROP_NTH_ASM_T 11(ante_tac o ∀_elim⌜last'⌝)
        THEN asm_rewrite_tac[size_list_rel_⌢_▷_thm] THEN ⇒_tac);
a(POP_ASM_T rewrite_thm_tac);
a(asm_rewrite_tac[list_rel_⌢_▷_thm,rel_combine_one_lemma]);
a(strip_asm_tac(list_∀_elim[⌜l2⌝,⌜last⌝,⌜c2⌝,⌜us⌝,⌜t2⌝]conjunct1_lemma1));
a(asm_rewrite_tac[]);
```

SML

```
(* *** Goal "1.2.2.4" *** *)
a(lemma_tac⌜¬ c2 dominates R_exist last ∧ ¬ c2 dominates R_exist last'⌝);
(* *** Goal "1.2.2.4.1" *** *)
a(swap_nth_asm_concl_tac 1 THEN fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "1.2.2.4.2" *** *)
a(asm_rewrite_tac[size_list_rel_⌒_▷_thm] THEN REPEAT ⇒_tac
        THEN fc_tac[rel_combine_null_lemma]);
a(DROP_NTH_ASM_T 13(ante_tac o ∀_elim⌜l2⌝)
        THEN DROP_NTH_ASM_T 12(fn _ => id_tac)
        THEN asm_rewrite_tac[size_list_rel_⌒_▷_thm] THEN ⇒_tac);
a(asm_rewrite_tac[list_rel_⌒_▷_thm,rel_combine_one_lemma]);
a(strip_asm_tac(list_∀_elim[⌜l1⌝,⌜last'⌝,⌜c2⌝,⌜us⌝,⌜t2⌝]conjunct1_lemma1));
a(strip_asm_tac(list_∀_elim[⌜l2⌝,⌜last⌝,⌜c2⌝,⌜us⌝,⌜t2⌝]conjunct1_lemma1));
a(asm_rewrite_tac[]);
val cleanTable_updateRows_lemma = save_pop_thm"cleanTable_updateRows_lemma";
```

HOL output

```
cleanTable_updateRows_lemma =
⊢ ∀ c1 c2 t1 t2 us
  • Dom (⋃ (Ran us))
        ⊆ {n|∃ c• c ∈ Snd (cleanColCons c2 t2) ∧ CS_posn c = n}
      ∧ ((RelCombine
              ((revealRow c2 t1)~ ⨾ us)
              (ListRel (TS_rows t1))
            ⨾ Graph (updateRow c2 (TS_class t2)))
          ▷ {x|isError x})
        ⨾ Graph destError
      = {}
      ∧ ((RelCombine
              ((revealRow c2 t2)~ ⨾ us)
              (ListRel (TS_rows t2))
            ⨾ Graph (updateRow c2 (TS_class t2)))
          ▷ {x|isError x})
        ⨾ Graph destError
      = {}
      ∧ cleanTable c1 t1 = cleanTable c1 t2
      ∧ c1 dominates c2
      ∧ c2 dominates TS_class t1
    ⇒ cleanTable
        c1
        (replaceRows
            t1
            (RelList
                (ListRel (TS_rows t1)
                    ⊕ (RelCombine
                        ((revealRow c2 t1)~ ⨾ us)
                        (ListRel (TS_rows t1))
                      ⨾ Graph (updateRow c2 (TS_class t1)))
                    ⨾ Graph destVal)))
      = cleanTable
        c1
        (replaceRows
            t2
            (RelList
                (ListRel (TS_rows t2)
                    ⊕ (RelCombine
                        ((revealRow c2 t2)~ ⨾ us)
                        (ListRel (TS_rows t2))
                      ⨾ Graph (updateRow c2 (TS_class t2)))
                    ⨾ Graph destVal)))
```

3.4 Proof of Conjunct 2

First we prove three auxiliary results.

SML

```
push_goal([],⌜∀ c1 c2 i s • (c1 dominates c2 ∧ tabExists c2 i s)
        ⇒ tabExists c1 i s⌝);
a(REPEAT strip_tac);
a(POP_ASM_T ante_tac THEN rewrite_tac[tabExists_def]);
a(REPEAT strip_tac);
(* *** Goal "1" *** *)
a(∃_tac⌜y⌝THEN asm_rewrite_tac[]);
(* *** Goal "2" *** *)
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "3" *** *)
a(fc_tac[dominates_trans]THEN asm_fc_tac[]);
(* *** Goal "4" *** *)
a(∃_tac⌜y'⌝THEN asm_rewrite_tac[]);
val tabExists_lemma1 = save_pop_thm"tabExists_lemma1";
```

HOL output

```
tabExists_lemma1 =
⊢ ∀ c1 c2 i s
  • c1 dominates c2 ∧ tabExists c2 i s ⇒ tabExists c1 i s
```

SML

```
push_goal([],⌜∀ c s1 s2
  • hideR (c, repState s1) = hideR (c, repState s2)
      ⇒ (∀ i
      • tabExists c i (repState s1)
          ⇒ cleanTable c (getTable i (repState s1))
              = cleanTable c (getTable i (repState s2)))⌝);
a(REPEAT strip_tac);
a(fc_tac[rewrite_rule[hide_eq_lemma]tabExists_cleanTable_lemma]THEN asm_fc_tac[]);
val tabExists_cleanTable_lemma1 = save_pop_thm"tabExists_cleanTable_lemma1";
```

HOL output

```
tabExists_cleanTable_lemma1 =
⊢ ∀ c s1 s2
  • hideR (c, repState s1) = hideR (c, repState s2)
      ⇒ (∀ i
      • tabExists c i (repState s1)
          ⇒ cleanTable c (getTable i (repState s1))
            = cleanTable c (getTable i (repState s2)))
```

SML

```
push_goal([],⌜∀ c t1 t2 •
        c dominates TS_class t1
        ∧ c dominates TS_class t2
        ∧ cleanTable c t1 = cleanTable c t2
        ⇒ colDefaults c t1 = colDefaults c t2⌝);
a(REPEAT strip_tac);
a(POP_ASM_T ante_tac THEN asm_rewrite_tac
        [cleanTable_def,get_spec⌜MkTableSpec⌝,tab_components] THEN strip_tac);
a(asm_rewrite_tac[ext_thm,colDefaults_def,visibleCols_def]);
val colDefaults_lemma = save_pop_thm"colDefaults_lemma";
```

HOL output

```
colDefaults_lemma = ⊢ ∀ c t1 t2
  • c dominates TS_class t1
        ∧ c dominates TS_class t2
        ∧ cleanTable c t1 = cleanTable c t2
      ⇒ colDefaults c t1 = colDefaults c t2
```

Now the proof of *conjunct2*.

SML

```
push_goal([],⌜∀ c1 c2 s1 s2 e
        • hideR (c1, repState s1) = hideR (c1, repState s2) ∧ c1 dominates c2
            ⇒ hideR (c1, Fst (updateStateR (c2, e, repState s1)))
              = hideR (c1, Fst (updateStateR (c2, e, repState s2)))⌝);
a(REPEAT strip_tac);
a(lemma_tac⌜hideR (c2, repState s1) = hideR (c2, repState s2)⌝
        THEN_LIST[fc_tac[rewrite_rule[hide_eq_lemma,secureHide_def]secureHide_lemma]
        THEN asm_fc_tac[],id_tac]);
a(LEMMA_T⌜e = (Fst e,Snd e)⌝ pure_once_asm_rewrite_thm_tac
        THEN_LIST[rewrite_tac[],rewrite_tac[updateStateR_def]]);
a(lemma_tac⌜(tabExists c2 (tabFromEffect (Fst e)) (repState s1)
                ⇒ tabExists c2 (tabFromEffect (Fst e)) (repState s2))
        ∧ (tabExists c2(tabFromEffect (Fst e)) (repState s2 )
                ⇒ tabExists c2(tabFromEffect (Fst e)) (repState s1))⌝);
(*
```

SML

```
*)
(* *** Goal "1" *** *)
a(strip_asm_tac(list_∀_elim[⌜c2⌝,⌜s1⌝,⌜s2⌝](rewrite_rule[hide_eq_lemma]
        tabExists_lemma)) THEN asm_rewrite_tac[]);
(* *** Goal "2" *** *)
(* invalid tables in s1 and s2 *)
a(strip_asm_tac(∀_elim ⌜Fst e⌝ query_type) THEN asm_rewrite_tac[]
        THEN cases_tac⌜¬ Snd e = []⌝ THEN asm_rewrite_tac[]);
(* *** Goal "3" *** *)
(* valid tables in s1 and s2 *)
a(strip_asm_tac (list_∀_elim[⌜c2⌝,⌜s1⌝,⌜s2⌝](rewrite_rule[hide_eq_lemma]
        tabExists_cleanTable_lemma)));
a(POP_ASM_T (strip_asm_tac o ∀_elim⌜tabFromEffect (Fst e)⌝));
(*
```

SML

```
*)
(* *** Goal "3.1" *** *)
(* table classes not dominated by c *)
a(EVERY[strip_asm_tac(∀_elim ⌜Fst e⌝ query_type),
        asm_rewrite_tac[],
        cases_tac⌜¬ Snd e = []⌝,
        asm_rewrite_tac[]]);
(* *** Goal "3.2" *** *)
(* table classes dominated by c *)
a(strip_asm_tac(list_∀_elim[⌜c1⌝,⌜c2⌝,⌜tabFromEffect (Fst e)⌝,⌜repState s1⌝]tabExists_lemma1));
a(strip_asm_tac (list_∀_elim[⌜c1⌝,⌜s1⌝,⌜s2⌝]tabExists_cleanTable_lemma1));
a(POP_ASM_T (strip_asm_tac o ∀_elim⌜tabFromEffect (Fst e)⌝));
a(EVERY[strip_asm_tac(∀_elim ⌜Fst e⌝ query_type),
        asm_rewrite_tac[],
        cases_tac⌜¬ Snd e = []⌝,
        asm_rewrite_tac[]]);
(*
```

SML

```
*)
(* 3 subgoals − Select automatically proven *)
(* *** Goal "3.2.1" *** *)
(* *** Insert *** *)
a(LIST_DROP_NTH_ASM_T[1,2,3,4,7](MAP_EVERY (fn _ => id_tac)));
a(POP_ASM_T (strip_asm_tac o rewrite_rule[isInsert_def]));
a(LIST_DROP_NTH_ASM_T[2,3,4,5,6,7](MAP_EVERY ante_tac)
        THEN POP_ASM_T rewrite_thm_tac);
a(rewrite_tac[destInsert_def,tabFromEffect_def,
        getTable_def,rewrite_rule[dom_def]tabExists_def]);
a(REPEAT strip_tac);
a(strip_asm_tac (pure_rewrite_rule[get_spec⌜isState⌝,get_spec⌜StateS⌝,↦_def,∩_def]
        (∀_elim⌜s1⌝isState_lemma)));
a(strip_asm_tac(list_∀_elim[⌜repState s1⌝,⌜Front (Fst i)⌝,⌜y''⌝]at_thm1));
a(strip_asm_tac (pure_rewrite_rule[get_spec⌜isState⌝,get_spec⌜StateS⌝,↦_def,∩_def]
        (∀_elim⌜s2⌝isState_lemma)));
a(strip_asm_tac(list_∀_elim[⌜repState s2⌝,⌜Front (Fst i)⌝,⌜y⌝]at_thm1));
a(LIST_DROP_NTH_ASM_T[7,8,9,10,11,12,13,15,16,17](MAP_EVERY ante_tac)
        THEN GET_NTH_ASM_T 4 rewrite_thm_tac THEN TOP_ASM_T rewrite_thm_tac);
(*
```

SML

```
|*)
|a(REPEAT ⇒_tac);
|a(DROP_NTH_ASM_T 16 ante_tac THEN DROP_NTH_ASM_T 13 ante_tac THEN
|        rewrite_tac[↔_def,get_spec⌜IdeL⌝,get_spec⌜DirectoryS⌝,∩_def,×_def,
|        get_spec⌜Universe⌝,rel_ext_clauses,get_spec⌜$ℙ⌝]THEN REPEAT ⇒_tac);
|a(asm_fc_tac[]);
|a(POP_ASM_T ante_tac THEN POP_ASM_T ante_tac THEN rewrite_tac[⇸_def,∩_def]
|        THEN REPEAT ⇒_tac);
|a(strip_asm_tac(list_∀_elim[⌜Dir_tables y⌝,⌜Last (Fst i)⌝,⌜y'⌝]at_thm1));
|a(strip_asm_tac(list_∀_elim[⌜Dir_tables y''⌝,⌜Last (Fst i)⌝,⌜y'''⌝]at_thm1));
|a(LIST_DROP_NTH_ASM_T[9,10,11,12](MAP_EVERY ante_tac)THEN asm_rewrite_tac[]);
|a(LIST_DROP_NTH_ASM_T[4,6,7,8](MAP_EVERY (fn _ => id_tac))
|        THEN REPEAT ⇒_tac);
|a(LEMMA_T⌜i = (Fst i,Snd i)⌝ pure_once_asm_rewrite_thm_tac
|        THEN_LIST[rewrite_tac[],asm_rewrite_tac[hideR_def,insertQuery_def,changeSpec_def]]);
|a(DROP_NTH_ASM_T 23(strip_asm_tac o rewrite_rule[hideR_def,rel_ext_clauses]));
|a(REPEAT ∀_tac);
|a(strip_asm_tac(list_∀_elim[⌜c2⌝,⌜y'''⌝,⌜y'⌝]colDefaults_lemma));
|a(TOP_ASM_T rewrite_thm_tac);
|a(cases_tac⌜¬ Elems (Map (MkRow c2 o colDefaults c2 y')
|                (Snd i)) ⊆ RowS⌝
|        THEN asm_rewrite_tac[]);
|a(POP_ASM_T (fn _ => id_tac) THEN REPEAT strip_tac);
|(*
```

SML

```
*)
(* *** Goal "3.2.1.1" *** *)
a(rewrite_tac[⊕_single]);
a(CASES_T⌜x = Front (Fst i)⌝asm_tac);
(* *** Goal "3.2.1.1.1" *** *)
a(DROP_NTH_ASM_T 3(strip_asm_tac o rewrite_rule
        [⊕_single,get_spec⌜MkDirectory⌝,dir_components]));
a(DROP_NTH_ASM_T 8(asm_tac o list_∀_elim[⌜Front(Fst i)⌝,⌜cleanDirectory c1 y''⌝]));
a(LEMMA_T⌜(∃ z
            • (c1 dominates Dir_exist z ∧ (Front (Fst i), z) ∈ repState s1)
               ∧ cleanDirectory c1 y'' = cleanDirectory c1 z)⌝asm_tac);
(* *** Goal "3.2.1.1.1.1" *** *)
a(∃_tac⌜y''⌝THEN asm_rewrite_tac[]);
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
(* *** Goal "3.2.1.1.1.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(lemma_tac⌜z' = y⌝);
(*
```

SML

```
*)
(* *** Goal "3.2.1.1.1.2.1" *** *)
a(DROP_NTH_ASM_T 26(asm_tac o rewrite_rule[functional_def]));
a(POP_ASM_T (strip_asm_tac o list_∀_elim[⌜Front (Fst i)⌝,⌜z'⌝,⌜y⌝]));
(* *** Goal "3.2.1.1.1.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac);
a(⇒_T (strip_asm_tac o rewrite_rule[cleanDirectory_def,
        get_spec⌜MkDirectory⌝,dir_components]));
a(∃_tac⌜MkDirectory
                        (Dir_tables y
                            ⊕ {(Last (Fst i),
                                replaceRows
                                    y'
                                    (TS_rows y'
                                        ⌢ Map
                                          (MkRow c2 o colDefaults c2 y')
                                          (Snd i)))})
                        (Dir_exist y)
                        (Dir_class y)⌝THEN asm_rewrite_tac[get_spec⌜MkDirectory⌝]);
a(lemma_tac⌜c1 dominates Dir_exist y⌝ THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(rewrite_tac[cleanDirectory_def]);
a(lemma_tac⌜c1 dominates Dir_class y''⌝ THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(asm_rewrite_tac[get_spec⌜MkDirectory⌝,dir_components]);
a(lemma_tac⌜c1 dominates Dir_class y⌝ THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(DROP_NTH_ASM_T 6 ante_tac THEN asm_rewrite_tac[rel_ext_clauses]THEN ⇒_tac);
a(rewrite_tac[⊕_single]);
a(REPEAT ∀_tac THEN ⇔_T asm_tac);
(*
```

SML

```
*)
(* *** Goal "3.2.1.1.1.2.2.1" *** *)
a(cases_tac⌜x' = Last (Fst i)⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.1.1.1.2.2.1.1" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜replaceRows y' (TS_rows y' ⌢ Map (MkRow c2 o colDefaults c2 y') (Snd i))⌝
        THEN asm_rewrite_tac[]);
a(EXTEND_PC_T1 "'mmp1" fc_tac[cleanTable_insertRows_lemma]
        THEN EXTEND_PC_T1 "'mmp1" asm_fc_tac[]);
a(DROP_NTH_ASM_T 7 (ante_tac o ∀_elim⌜Snd i⌝) THEN asm_rewrite_tac[]);
(* *** Goal "3.2.1.1.1.2.2.1.2" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(DROP_NTH_ASM_T 4 (asm_tac o list_∀_elim[⌜x'⌝,⌜cleanTable c1 z''⌝]));
a(LEMMA_T⌜∃ z • (x', z) ∈ Dir_tables y''
                ∧ cleanTable c1 z'' = cleanTable c1 z⌝asm_tac);
(* *** Goal "3.2.1.1.1.2.2.1.2.1" *** *)
a(∃_tac⌜z''⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.1.1.1.2.2.1.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(∃_tac⌜z'''⌝THEN asm_rewrite_tac[]);
```

SML

```
(* *** Goal "3.2.1.1.1.2.2.2" *** *)
a(cases_tac⌜x' = Last (Fst i)⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.1.1.1.2.2.2.1" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜replaceRows y''' (TS_rows y''' ⌢ Map (MkRow c2 o colDefaults c2 y''') (Snd i))⌝
        THEN asm_rewrite_tac[]);
a(EXTEND_PC_T1 "'mmp1" fc_tac[cleanTable_insertRows_lemma]
        THEN EXTEND_PC_T1 "'mmp1" asm_fc_tac[]);
a(DROP_NTH_ASM_T 7 (ante_tac o ∀_elim⌜Snd i⌝) THEN asm_rewrite_tac[]
        THEN ⇒_T rewrite_thm_tac);
(* *** Goal "3.2.1.1.1.2.2.2.2" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜z''⌝THEN asm_rewrite_tac[]);
(*
```

SML

```
*)
(* *** Goal "3.2.1.1.2" *** *)
a(DROP_NTH_ASM_T 3(strip_asm_tac o rewrite_rule
        [⊕_single,get_spec⌜MkDirectory⌝,dir_components]));
a(DROP_NTH_ASM_T 6(asm_tac o list_∀_elim[⌜x⌝,⌜cleanDirectory c1 z⌝]));
a(LEMMA_T⌜∃ z'
            • (c1 dominates Dir_exist z' ∧ (x, z') ∈ repState s1)
                ∧ cleanDirectory c1 z = cleanDirectory c1 z'⌝asm_tac);
(* *** Goal "3.2.1.1.2.1" *** *)
a(∃_tac⌜z⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.1.1.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(∃_tac⌜z'⌝THEN asm_rewrite_tac[]);
(*
```

SML

```
*)
(* *** Goal "3.2.1.2" *** *)
a(rewrite_tac[⊕_single]);
a(CASES_T⌜x = Front (Fst i)⌝asm_tac);
(* *** Goal "3.2.1.2.1" *** *)
a(DROP_NTH_ASM_T 3(strip_asm_tac o rewrite_rule
        [⊕_single,get_spec⌜MkDirectory⌝,dir_components]));
a(DROP_NTH_ASM_T 8(asm_tac o list_∀_elim[⌜Front(Fst i)⌝,⌜cleanDirectory c1 y''⌝]));
a(LEMMA_T⌜(∃ z
            • (c1 dominates Dir_exist z ∧ (Front (Fst i), z) ∈ repState s1)
                ∧ cleanDirectory c1 y'' = cleanDirectory c1 z)⌝asm_tac);
(* *** Goal "3.2.1.2.1.1" *** *)
a(∃_tac⌜y''⌝THEN asm_rewrite_tac[]);
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
(* *** Goal "3.2.1.2.1.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(lemma_tac⌜z' = y⌝);
(*
```

SML

```
*)
(* *** Goal "3.2.1.2.1.2.1" *** *)
a(DROP_NTH_ASM_T 26(asm_tac o rewrite_rule[functional_def]));
a(POP_ASM_T (strip_asm_tac o list_∀_elim[⌜Front (Fst i)⌝,⌜z'⌝,⌜y⌝]));
(* *** Goal "3.2.1.2.1.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac);
a(⇒_T (strip_asm_tac o rewrite_rule[cleanDirectory_def,
        get_spec⌜MkDirectory⌝,dir_components]));
a(∃_tac⌜MkDirectory
                        (Dir_tables y''
                            ⊕ {(Last (Fst i),
                                replaceRows
                                    y'''
                                    (TS_rows y'''
                                        ⌢ Map
                                          (MkRow c2 o colDefaults c2 y''')
                                          (Snd i)))})
                        (Dir_exist y'')
                        (Dir_class y'')⌝THEN asm_rewrite_tac[get_spec⌜MkDirectory⌝]);
a(lemma_tac⌜c1 dominates Dir_exist y⌝ THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(rewrite_tac[cleanDirectory_def]);
a(lemma_tac⌜c1 dominates Dir_class y''⌝ THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(asm_rewrite_tac[get_spec⌜MkDirectory⌝,dir_components]);
a(lemma_tac⌜c1 dominates Dir_class y⌝ THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(DROP_NTH_ASM_T 6 ante_tac THEN asm_rewrite_tac[rel_ext_clauses]THEN ⇒_tac);
a(rewrite_tac[⊕_single]);
a(REPEAT ∀_tac THEN ⇔_T asm_tac);
(*
```

SML

```
*)
(* *** Goal "3.2.1.2.1.2.2.1" *** *)
a(cases_tac⌜x' = Last (Fst i)⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.1.2.1.2.2.1.1" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜replaceRows y''' (TS_rows y''' ⌢ Map (MkRow c2 o colDefaults c2 y''') (Snd i))⌝
        THEN asm_rewrite_tac[]);
a(EXTEND_PC_T1 "'mmp1" fc_tac[cleanTable_insertRows_lemma]
        THEN EXTEND_PC_T1 "'mmp1" asm_fc_tac[]);
a(DROP_NTH_ASM_T 7 (ante_tac o ∀_elim⌜Snd i⌝) THEN asm_rewrite_tac[]
        THEN ⇒_T rewrite_thm_tac);
(* *** Goal "3.2.1.2.1.2.2.1.2" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜z''⌝THEN asm_rewrite_tac[]);
(*
```

SML

```
*)
(* *** Goal "3.2.1.2.1.2.2.2" *** *)
a(cases_tac⌜x' = Last (Fst i)⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.1.2.1.2.2.2.1" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜replaceRows y' (TS_rows y' ⌢ Map (MkRow c2 o colDefaults c2 y') (Snd i))⌝
        THEN asm_rewrite_tac[]);
a(EXTEND_PC_T1 "'mmp1" fc_tac[cleanTable_insertRows_lemma]
        THEN EXTEND_PC_T1 "'mmp1" asm_fc_tac[]);
a(DROP_NTH_ASM_T 7 (ante_tac o ∀_elim⌜Snd i⌝) THEN asm_rewrite_tac[]);
(* *** Goal "3.2.1.2.1.2.2.2.2" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(DROP_NTH_ASM_T 4 (asm_tac o list_∀_elim[⌜x'⌝,⌜cleanTable c1 z''⌝]));
a(LEMMA_T⌜∃ z • (x', z) ∈ Dir_tables y''
                ∧ cleanTable c1 z'' = cleanTable c1 z⌝asm_tac);
(* *** Goal "3.2.1.2.1.2.2.2.2.1" *** *)
a(∃_tac⌜z''⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.1.2.1.2.2.2.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(∃_tac⌜z'''⌝THEN asm_rewrite_tac[]);
(*
```

SML

```
*)
(* *** Goal "3.2.1.2.2" *** *)
a(DROP_NTH_ASM_T 3(strip_asm_tac o rewrite_rule
        [⊕_single,get_spec⌜MkDirectory⌝,dir_components]));
a(DROP_NTH_ASM_T 6(asm_tac o list_∀_elim[⌜x⌝,⌜cleanDirectory c1 z⌝]));
a(LEMMA_T⌜∃ z'
            • (c1 dominates Dir_exist z' ∧ (x, z') ∈ repState s2)
              ∧ cleanDirectory c1 z = cleanDirectory c1 z'⌝asm_tac);
(* *** Goal "3.2.1.2.2.1" *** *)
a(∃_tac⌜z⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.1.2.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(∃_tac⌜z'⌝THEN asm_rewrite_tac[]);
(*
```

SML

```
*)
(* *** Goal "3.2.2" *** *)
(* *** Delete *** *)
a(LIST_DROP_NTH_ASM_T[1,2,3,4,7](MAP_EVERY (fn _ => id_tac)));
a(POP_ASM_T (strip_asm_tac o rewrite_rule[isDelete_def]));
a(LIST_DROP_NTH_ASM_T[2,3,4,5,6,7](MAP_EVERY ante_tac)
        THEN POP_ASM_T rewrite_thm_tac);
a(rewrite_tac[destInsert_def,tabFromEffect_def,
        getTable_def,rewrite_rule[dom_def]tabExists_def]);
a(REPEAT strip_tac);
a(strip_asm_tac (pure_rewrite_rule[get_spec⌜isState⌝,get_spec⌜StateS⌝,↦_def,∩_def]
        (∀_elim⌜s1⌝isState_lemma)));
a(strip_asm_tac(list_∀_elim[⌜repState s1⌝,⌜Front (Fst d)⌝,⌜y''⌝]at_thm1));
a(strip_asm_tac (pure_rewrite_rule[get_spec⌜isState⌝,get_spec⌜StateS⌝,↦_def,∩_def]
        (∀_elim⌜s2⌝isState_lemma)));
a(strip_asm_tac(list_∀_elim[⌜repState s2⌝,⌜Front (Fst d)⌝,⌜y⌝]at_thm1));
a(LIST_DROP_NTH_ASM_T[7,8,9,10,11,12,13,15,16,17](MAP_EVERY ante_tac)
        THEN GET_NTH_ASM_T 4 rewrite_thm_tac THEN TOP_ASM_T rewrite_thm_tac);
(*
```

SML

```
*)
a(REPEAT ⇒_tac);
a(DROP_NTH_ASM_T 16 ante_tac THEN DROP_NTH_ASM_T 13 ante_tac THEN
        rewrite_tac[↔_def,get_spec⌜IdeL⌝,get_spec⌜DirectoryS⌝,∩_def,×_def,
        get_spec⌜Universe⌝,rel_ext_clauses,get_spec⌜$ℙ⌝]THEN REPEAT ⇒_tac);
a(asm_fc_tac[]);
a(POP_ASM_T ante_tac THEN POP_ASM_T ante_tac THEN rewrite_tac[↦_def,∩_def]
        THEN REPEAT ⇒_tac);
a(strip_asm_tac(list_∀_elim[⌜Dir_tables y⌝,⌜Last (Fst d)⌝,⌜y'⌝]at_thm1));
a(strip_asm_tac(list_∀_elim[⌜Dir_tables y''⌝,⌜Last (Fst d)⌝,⌜y'''⌝]at_thm1));
a(LIST_DROP_NTH_ASM_T[9,10,11,12](MAP_EVERY ante_tac)THEN asm_rewrite_tac[]);
a(LIST_DROP_NTH_ASM_T[4,6,7,8](MAP_EVERY (fn _ => id_tac))
        THEN REPEAT ⇒_tac);
a(LEMMA_T⌜d = (Fst d,Snd d)⌝ pure_once_asm_rewrite_thm_tac
        THEN_LIST[rewrite_tac[],asm_rewrite_tac[hideR_def,deleteQuery_def,changeSpec_def]]);
a(DROP_NTH_ASM_T 23(strip_asm_tac o rewrite_rule[hideR_def,rel_ext_clauses]));
a(REPEAT strip_tac);
(*
```

SML

```
*)
(* *** Goal "3.2.2.1" *** *)
a(rewrite_tac[⊕_single]);
a(CASES_T⌜x = Front (Fst d)⌝asm_tac);
(* *** Goal "3.2.2.1.1" *** *)
a(DROP_NTH_ASM_T 3(strip_asm_tac o rewrite_rule
        [⊕_single,get_spec⌜MkDirectory⌝,dir_components]));
a(DROP_NTH_ASM_T 7(asm_tac o list_∀_elim[⌜Front(Fst d)⌝,⌜cleanDirectory c₁ y''⌝]));
a(LEMMA_T⌜(∃ z
            • (c₁ dominates Dir_exist z ∧ (Front (Fst d), z) ∈ repState s₁)
              ∧ cleanDirectory c₁ y'' = cleanDirectory c₁ z)⌝asm_tac);
(* *** Goal "3.2.2.1.1.1" *** *)
a(∃_tac⌜y''⌝THEN asm_rewrite_tac[]);
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
(* *** Goal "3.2.2.1.1.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(lemma_tac⌜z' = y⌝);
(*
```

SML

```
|*)
|(* *** Goal "3.2.2.1.1.2.1" *** *)
|a(DROP_NTH_ASM_T 25(asm_tac o rewrite_rule[functional_def]));
|a(POP_ASM_T (strip_asm_tac o list_∀_elim[⌜Front (Fst d)⌝,⌜z'⌝,⌜y⌝]));
|(* *** Goal "3.2.2.1.1.2.2" *** *)
|a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac);
|a(⇒_T (strip_asm_tac o rewrite_rule[cleanDirectory_def,
|        get_spec⌜MkDirectory⌝,dir_components]));
|(*
```

SML

```
|*)
|a(∃_tac⌜MkDirectory(Dir_tables y
|                              ⊕ {(Last (Fst d),
|                                 replaceRows
|                                     y'
|                                     (Extract
|                                         (1 .. # (TS_rows y')
|                                             \ revealRow c2 y' Image Snd d
|                                               ∩ {i
|                                               |R_exist (Nth (TS_rows y') i)
|                                                   = c2})
|                                         (TS_rows y'))))})
|                         (Dir_exist y)
|                         (Dir_class y)⌝THEN asm_rewrite_tac[get_spec⌜MkDirectory⌝]);
|(*
```

SML

```
|*)
|a(lemma_tac⌜c1 dominates Dir_exist y⌝ THEN_LIST
|        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
|a(rewrite_tac[cleanDirectory_def]);
|a(lemma_tac⌜c1 dominates Dir_class y''⌝ THEN_LIST
|        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
|a(asm_rewrite_tac[get_spec⌜MkDirectory⌝,dir_components]);
|a(lemma_tac⌜c1 dominates Dir_class y⌝ THEN_LIST
|        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
|a(DROP_NTH_ASM_T 6 ante_tac THEN asm_rewrite_tac[rel_ext_clauses]THEN ⇒_tac);
|a(rewrite_tac[⊕_single]);
|a(REPEAT ∀_tac THEN ⇔_T asm_tac);
|(*
```

SML

```
*)
(* *** Goal "3.2.2.1.1.2.2.1" *** *)
a(cases_tac⌜x' = Last (Fst d)⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.2.1.1.2.2.1.1" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜replaceRows y'
                    (Extract
                        (1 .. # (TS_rows y')
                            \ revealRow c2 y' Image Snd d
                              ∩ {i|R_exist (Nth (TS_rows y') i) = c2})
                        (TS_rows y'))⌝ THEN asm_rewrite_tac[]);
a(EXTEND_PC_T1 "'mmp1" fc_tac[cleanTable_deleteRows_lemma]
        THEN EXTEND_PC_T1 "'mmp1" asm_fc_tac[]);
a(spec_nth_asm_tac 7 ⌜Snd d⌝);
(* *** Goal "3.2.2.1.1.2.2.1.2" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(DROP_NTH_ASM_T 4 (asm_tac o list_∀_elim[⌜x'⌝,⌜cleanTable c1 z''⌝]));
a(LEMMA_T⌜∃ z • (x', z) ∈ Dir_tables y''
                ∧ cleanTable c1 z'' = cleanTable c1 z⌝asm_tac);
(*
```

SML

```
|*)
|(* *** Goal "3.2.2.1.1.2.2.1.2.1" *** *)
|a(∃_tac⌜z''⌝THEN asm_rewrite_tac[]);
|(* *** Goal "3.2.2.1.1.2.2.1.2.2" *** *)
|a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
|a(∃_tac⌜z'''⌝THEN asm_rewrite_tac[]);
|(* *** Goal "3.2.2.1.1.2.2.2" *** *)
|a(cases_tac⌜x' = Last (Fst d)⌝THEN asm_rewrite_tac[]);
|(* *** Goal "3.2.2.1.1.2.2.2.1" *** *)
|a(DROP_NTH_ASM_T 2 strip_asm_tac);
|a(∃_tac⌜replaceRows y'''
|                  (Extract
|                     (1 .. # (TS_rows y''')
|                        \ revealRow c_2 y''' Image Snd d
|                          ∩ {i|R_exist (Nth (TS_rows y''') i) = c_2})
|                     (TS_rows y'''))⌝ THEN asm_rewrite_tac[]);
|a(EXTEND_PC_T1 "'mmp1" fc_tac[cleanTable_deleteRows_lemma]
|       THEN EXTEND_PC_T1 "'mmp1" asm_fc_tac[]);
|a(spec_nth_asm_tac 7 ⌜Snd d⌝THEN asm_rewrite_tac[]);
|(* *** Goal "3.2.2.1.1.2.2.2.2" *** *)
|a(DROP_NTH_ASM_T 2 strip_asm_tac);
|a(∃_tac⌜z''⌝THEN asm_rewrite_tac[]);
|(*
```

SML

```
|*)
|(* *** Goal "3.2.2.1.2" *** *)
|a(DROP_NTH_ASM_T 3(strip_asm_tac o rewrite_rule
|       [⊕_single,get_spec⌜MkDirectory⌝,dir_components]));
|a(DROP_NTH_ASM_T 5(asm_tac o list_∀_elim[⌜x⌝,⌜cleanDirectory c_1 z⌝]));
|a(LEMMA_T⌜∃ z'
|            • (c_1 dominates Dir_exist z' ∧ (x, z') ∈ repState s_1)
|              ∧ cleanDirectory c_1 z = cleanDirectory c_1 z'⌝asm_tac);
|(* *** Goal "3.2.2.1.2.1" *** *)
|a(∃_tac⌜z⌝THEN asm_rewrite_tac[]);
|(* *** Goal "3.2.2.1.2.2" *** *)
|a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
|a(∃_tac⌜z'⌝THEN asm_rewrite_tac[]);
|(*
```

SML

```
*)
(* *** Goal "3.2.2.2" *** *)
a(rewrite_tac[⊕_single]);
a(CASES_T⌜x = Front (Fst d)⌝asm_tac);
(* *** Goal "3.2.2.2.1" *** *)
a(DROP_NTH_ASM_T 3(strip_asm_tac o rewrite_rule
        [⊕_single,get_spec⌜MkDirectory⌝,dir_components]));
a(DROP_NTH_ASM_T 7(asm_tac o list_∀_elim[⌜Front(Fst d)⌝,⌜cleanDirectory c1 y''⌝]));
a(LEMMA_T⌜(∃ z
            • (c1 dominates Dir_exist z ∧ (Front (Fst d), z) ∈ repState s1)
              ∧ cleanDirectory c1 y'' = cleanDirectory c1 z)⌝asm_tac);
(* *** Goal "3.2.2.2.1.1" *** *)
a(∃_tac⌜y''⌝THEN asm_rewrite_tac[]);
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
(* *** Goal "3.2.2.2.1.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(lemma_tac⌜z' = y⌝);
(*
```

SML

```
*)
(* *** Goal "3.2.2.2.1.2.1" *** *)
a(DROP_NTH_ASM_T 25(asm_tac o rewrite_rule[functional_def]));
a(POP_ASM_T (strip_asm_tac o list_∀_elim[⌜Front (Fst d)⌝,⌜z'⌝,⌜y⌝]));
(* *** Goal "3.2.2.2.1.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac);
a(⇒_T (strip_asm_tac o rewrite_rule[cleanDirectory_def,
        get_spec⌜MkDirectory⌝,dir_components]));
a(∃_tac⌜MkDirectory (Dir_tables y''
                              ⊕ {(Last (Fst d),
                                  replaceRows
                                      y'''
                                      (Extract
                                          (1 .. # (TS_rows y''')
                                              \ revealRow c2 y''' Image Snd d
                                                ∩ {i
                                                |R_exist (Nth (TS_rows y''') i)
                                                    = c2})
                                          (TS_rows y''')))})
                        (Dir_exist y'')
                        (Dir_class y'')⌝THEN asm_rewrite_tac[get_spec⌜MkDirectory⌝]);
(*
```

SML

```
*)
a(lemma_tac⌜c1 dominates Dir_exist y⌝ THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(rewrite_tac[cleanDirectory_def]);
a(lemma_tac⌜c1 dominates Dir_class y''⌝ THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(asm_rewrite_tac[get_spec⌜MkDirectory⌝,dir_components]);
a(lemma_tac⌜c1 dominates Dir_class y⌝ THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(DROP_NTH_ASM_T 6 ante_tac THEN asm_rewrite_tac[rel_ext_clauses]THEN ⇒_tac);
a(rewrite_tac[⊕_single]);
a(REPEAT ∀_tac THEN ⇔_T asm_tac);
(*
```

SML

```
*)
(* *** Goal "3.2.2.2.1.2.2.1" *** *)
a(cases_tac⌜x' = Last (Fst d)⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.2.2.1.2.2.1.1" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜replaceRows y'''
                (Extract
                    (1 .. # (TS_rows y''')
                        \ revealRow c_2 y''' Image Snd d
                          ∩ {i|R_exist (Nth (TS_rows y''') i) = c_2})
                    (TS_rows y'''))⌝ THEN asm_rewrite_tac[]);
a(EXTEND_PC_T1 "'mmp1" fc_tac[cleanTable_deleteRows_lemma]
        THEN EXTEND_PC_T1 "'mmp1" asm_fc_tac[]);
a(spec_nth_asm_tac 7 ⌜Snd d⌝ THEN asm_rewrite_tac[]);
(* *** Goal "3.2.2.2.1.2.2.1.2" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜z''⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.2.2.1.2.2.2" *** *)
a(cases_tac⌜x' = Last (Fst d)⌝THEN asm_rewrite_tac[]);
(*
```

SML

```
*)
(* *** Goal "3.2.2.2.1.2.2.2.1" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜replaceRows y'
                    (Extract
                        (1 .. # (TS_rows y')
                            \ revealRow c2 y' Image Snd d
                              ∩ {i|R_exist (Nth (TS_rows y') i) = c2})
                        (TS_rows y'))⌝ THEN asm_rewrite_tac[]);
a(EXTEND_PC_T1 "'mmp1" fc_tac[cleanTable_deleteRows_lemma]
        THEN EXTEND_PC_T1 "'mmp1" asm_fc_tac[]);
a(spec_nth_asm_tac 7 ⌜Snd d⌝);
(* *** Goal "3.2.2.2.1.2.2.2.2" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(DROP_NTH_ASM_T 4 (asm_tac o list_∀_elim[⌜x'⌝,⌜cleanTable c1 z''⌝]));
a(LEMMA_T⌜∃ z • (x', z) ∈ Dir_tables y''
                ∧ cleanTable c1 z'' = cleanTable c1 z⌝asm_tac);
(* *** Goal "3.2.2.2.1.2.2.2.2.1" *** *)
a(∃_tac⌜z''⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.2.2.1.2.2.2.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(∃_tac⌜z'''⌝THEN asm_rewrite_tac[]);
(*
```

SML

```
*)
(* *** Goal "3.2.2.2.2" *** *)
a(DROP_NTH_ASM_T 3(strip_asm_tac o rewrite_rule
        [⊕_single,get_spec⌜MkDirectory⌝,dir_components]));
a(DROP_NTH_ASM_T 5(asm_tac o list_∀_elim[⌜x⌝,⌜cleanDirectory c1 z⌝]));
a(LEMMA_T⌜∃ z'
            • (c1 dominates Dir_exist z' ∧ (x, z') ∈ repState s2)
                ∧ cleanDirectory c1 z = cleanDirectory c1 z'⌝asm_tac);
(* *** Goal "3.2.2.2.2.1" *** *)
a(∃_tac⌜z⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.2.2.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(∃_tac⌜z'⌝THEN asm_rewrite_tac[]);
(*
```

SML

```
*)
(* *** Goal "3.2.3" *** *)
(* *** Update *** *)
a(LIST_DROP_NTH_ASM_T[1,2,3,4,7](MAP_EVERY (fn _ => id_tac)));
a(POP_ASM_T (strip_asm_tac o rewrite_rule[isUpdate_def]));
a(LIST_DROP_NTH_ASM_T[2,3,4,5,6,7](MAP_EVERY ante_tac)
        THEN POP_ASM_T rewrite_thm_tac);
a(rewrite_tac[destInsert_def,tabFromEffect_def,
        getTable_def,rewrite_rule[dom_def]tabExists_def]);
a(REPEAT strip_tac);
a(strip_asm_tac (pure_rewrite_rule[get_spec⌜isState⌝,get_spec⌜StateS⌝,↦_def,∩_def]
        (∀_elim⌜s1⌝isState_lemma)));
a(strip_asm_tac(list_∀_elim[⌜repState s1⌝,⌜Front (Fst u)⌝,⌜y''⌝]at_thm1));
a(strip_asm_tac (pure_rewrite_rule[get_spec⌜isState⌝,get_spec⌜StateS⌝,↦_def,∩_def]
        (∀_elim⌜s2⌝isState_lemma)));
a(strip_asm_tac(list_∀_elim[⌜repState s2⌝,⌜Front (Fst u)⌝,⌜y⌝]at_thm1));
a(LIST_DROP_NTH_ASM_T[7,8,9,10,11,12,13,15,16,17](MAP_EVERY ante_tac)
        THEN GET_NTH_ASM_T 4 rewrite_thm_tac THEN TOP_ASM_T rewrite_thm_tac);
(*
```

SML

```
*)
a(REPEAT ⇒_tac);
a(DROP_NTH_ASM_T 16 ante_tac THEN DROP_NTH_ASM_T 13 ante_tac THEN
        rewrite_tac[↔_def,get_spec⌜IdeL⌝,get_spec⌜DirectoryS⌝,∩_def,×_def,
        get_spec⌜Universe⌝,rel_ext_clauses,get_spec⌜$ℙ⌝]THEN REPEAT ⇒_tac);
a(asm_fc_tac[]);
a(POP_ASM_T ante_tac THEN POP_ASM_T ante_tac THEN rewrite_tac[↦_def,∩_def]
        THEN REPEAT ⇒_tac);
a(strip_asm_tac(list_∀_elim[⌜Dir_tables y⌝,⌜Last (Fst u)⌝,⌜y'⌝]at_thm1));
a(strip_asm_tac(list_∀_elim[⌜Dir_tables y''⌝,⌜Last (Fst u)⌝,⌜y'''⌝]at_thm1));
a(LIST_DROP_NTH_ASM_T[9,10,11,12](MAP_EVERY ante_tac)THEN asm_rewrite_tac[]);
a(LIST_DROP_NTH_ASM_T[4,6,7,8](MAP_EVERY (fn _ => id_tac))
        THEN REPEAT ⇒_tac);
(*
```

```
SML
|*)
|a(LEMMA_T⌜hideR (c1, Fst (updateQuery (c2, u, repState s1, y'''))) 
|        = hideR(c1, Fst (updateQuery (c2, u, repState s2, y')))⌝
|        rewrite_thm_tac);
|a(DROP_NTH_ASM_T 4 ante_tac THEN
|        asm_rewrite_tac[cleanTable_def,get_spec⌜MkTableSpec⌝,tab_components]);
|a strip_tac;
|a(DROP_NTH_ASM_T 27 (asm_tac o rewrite_rule[hideR_def]));
|a(LEMMA_T⌜u = (Fst u,Snd u)⌝ pure_once_asm_rewrite_thm_tac THEN_LIST[rewrite_tac[],
|        asm_rewrite_tac[hideR_def,updateQuery_def,changeSpec_def,visibleCols_def]]);
|a(conv_tac (MAP_C let_conv));
|(*
```

```
SML
|*)
|a(cases_tac⌜¬ Snd u ∈ Functional⌝ THEN
|        cases_tac⌜¬ Dom (⋃ (Ran (Snd u)))
|                        ⊆ {n
|                        |∃ c
|                          • c ∈ Snd (cleanColCons c2 y') ∧ CS_posn c = n}⌝
|        THEN asm_rewrite_tac[]);
|a(ante_tac (rewrite_rule[visibleCols_def]
|        (list_∀_elim[⌜c2⌝,⌜y'⌝,⌜y'''⌝,⌜Snd u⌝]cleanRows_errors_or_vals_lemma))
|        THEN asm_rewrite_tac[] THEN REPEAT strip_tac THEN asm_rewrite_tac[]);
|a(DROP_NTH_ASM_T 5(asm_tac o rewrite_rule[rel_ext_clauses]));
|a(rewrite_tac[rel_ext_clauses]THEN REPEAT strip_tac);
|(*
```

SML

```
*)
(* *** Goal "3.2.3.1" *** *)
a(rewrite_tac[⊕_single]);
a(CASES_T⌜x = Front (Fst u)⌝asm_tac);
(* *** Goal "3.2.3.1.1" *** *)
a(DROP_NTH_ASM_T 3(strip_asm_tac o rewrite_rule
        [⊕_single,get_spec⌜MkDirectory⌝,dir_components]));
a(DROP_NTH_ASM_T 7(asm_tac o list_∀_elim[⌜Front(Fst u)⌝,⌜cleanDirectory c1 y''⌝]));
a(LEMMA_T⌜(∃ z
            • (c1 dominates Dir_exist z ∧ (Front (Fst u), z) ∈ repState s1)
              ∧ cleanDirectory c1 y'' = cleanDirectory c1 z)⌝asm_tac);
(* *** Goal "3.2.3.1.1.1" *** *)
a(∃_tac⌜y''⌝THEN asm_rewrite_tac[]);
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
(* *** Goal "3.2.3.1.1.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(lemma_tac⌜z' = y⌝);
(* *** Goal "3.2.3.1.1.2.1" *** *)
a(DROP_NTH_ASM_T 33(asm_tac o rewrite_rule[functional_def]));
a(POP_ASM_T (strip_asm_tac o list_∀_elim[⌜Front (Fst u)⌝,⌜z'⌝,⌜y⌝]));
(*
```

SML

```
|*)
|(* *** Goal "3.2.3.1.1.2.2" *** *)
|a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac);
|a(⇒_T (strip_asm_tac o rewrite_rule[cleanDirectory_def,
|        get_spec⌜MkDirectory⌝,dir_components]));
|a(∃_tac⌜MkDirectory(Dir_tables y
|                              ⊕ {(Last (Fst u),
|                                  replaceRows
|                                      y'
|                                      (RelList
|                                          (ListRel (TS_rows y')
|                                              ⊕ (RelCombine
|                                                  ((revealRow c2 y')~
|                                                      ⨾ Snd u)
|                                                  (ListRel (TS_rows y'))
|                                                ⨾ Graph
|                                                  (updateRow c2 (TS_class y')))
|                                              ⨾ Graph destVal)))})
|                        (Dir_exist y)
|                        (Dir_class y)⌝THEN asm_rewrite_tac[get_spec⌜MkDirectory⌝]);
|(*
```

SML

```
|*)
|a(lemma_tac⌜c1 dominates Dir_exist y⌝ THEN_LIST
|        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
|a(rewrite_tac[cleanDirectory_def]);
|a(lemma_tac⌜c1 dominates Dir_class y''⌝ THEN_LIST
|        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
|a(asm_rewrite_tac[get_spec⌜MkDirectory⌝,dir_components]);
|a(lemma_tac⌜c1 dominates Dir_class y⌝ THEN_LIST
|        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
|a(DROP_NTH_ASM_T 6 ante_tac THEN asm_rewrite_tac[rel_ext_clauses]THEN ⇒_tac);
|a(rewrite_tac[⊕_single]);
|a(REPEAT ∀_tac THEN ⇔_T asm_tac);
|(*
```

SML

```
*)
(* *** Goal "3.2.3.1.1.2.2.1" *** *)
a(cases_tac⌜x' = Last (Fst u)⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.3.1.1.2.2.1.1" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜replaceRows
                  y'
                  (RelList
                      (ListRel (TS_rows y')
                          ⊕ (RelCombine
                              ((revealRow c2 y')~ ⨾ Snd u)
                              (ListRel (TS_rows y'))
                            ⨾ Graph (updateRow c2 (TS_class y')))
                          ⨾ Graph destVal))⌝ THEN asm_rewrite_tac[]);
a(ante_tac(list_∀_elim[⌜c1⌝,⌜c2⌝,⌜y'''⌝,⌜y'⌝,⌜Snd u⌝]cleanTable_updateRows_lemma)
        THEN asm_rewrite_tac[]);
(* *** Goal "3.2.3.1.1.2.2.1.2" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(DROP_NTH_ASM_T 4 (asm_tac o list_∀_elim[⌜x'⌝,⌜cleanTable c1 z''⌝]));
a(LEMMA_T⌜∃ z • (x', z) ∈ Dir_tables y''
                ∧ cleanTable c1 z'' = cleanTable c1 z⌝asm_tac);
(*
```

SML

```
*)
(* *** Goal "3.2.3.1.1.2.2.1.2.1" *** *)
a(∃_tac⌜z''⌝ THEN asm_rewrite_tac[]);
(* *** Goal "3.2.3.1.1.2.2.1.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(∃_tac⌜z'''⌝ THEN asm_rewrite_tac[]);
(* *** Goal "3.2.3.1.1.2.2.2" *** *)
a(cases_tac⌜x' = Last (Fst u)⌝ THEN asm_rewrite_tac[]);
(* *** Goal "3.2.3.1.1.2.2.2.1" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜replaceRows
                  y'''
                  (RelList
                      (ListRel (TS_rows y''')
                          ⊕ (RelCombine
                              ((revealRow c2 y''')~ ⨾ Snd u)
                              (ListRel (TS_rows y'''))
                            ⨾ Graph (updateRow c2 (TS_class y')))
                          ⨾ Graph destVal))⌝ THEN asm_rewrite_tac[]);
a(ante_tac(list_∀_elim[⌜c1⌝,⌜c2⌝,⌜y'''⌝,⌜y'⌝,⌜Snd u⌝]cleanTable_updateRows_lemma)
        THEN asm_rewrite_tac[] THEN ⇒_T rewrite_thm_tac);
(* *** Goal "3.2.3.1.1.2.2.2.2" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜z''⌝ THEN asm_rewrite_tac[]);
(*
```

SML

```
*)
(* *** Goal "3.2.3.1.2" *** *)
a(DROP_NTH_ASM_T 3(strip_asm_tac o rewrite_rule
        [⊕_single,get_spec⌜MkDirectory⌝,dir_components]));
a(DROP_NTH_ASM_T 5(asm_tac o list_∀_elim[⌜x⌝,⌜cleanDirectory c1 z⌝]));
a(LEMMA_T⌜∃ z'
            • (c1 dominates Dir_exist z' ∧ (x, z') ∈ repState s1)
                ∧ cleanDirectory c1 z = cleanDirectory c1 z'⌝asm_tac);
(* *** Goal "3.2.3.1.2.1" *** *)
a(∃_tac⌜z⌝ THEN asm_rewrite_tac[]);
(* *** Goal "3.2.3.1.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(∃_tac⌜z'⌝ THEN asm_rewrite_tac[]);
(*
```

SML

```
*)
(* *** Goal "3.2.3.2" *** *)
a(rewrite_tac[⊕_single]);
a(CASES_T⌜x = Front (Fst u)⌝asm_tac);
(* *** Goal "3.2.3.2.1" *** *)
a(DROP_NTH_ASM_T 3(strip_asm_tac o rewrite_rule
        [⊕_single,get_spec⌜MkDirectory⌝,dir_components]));
a(DROP_NTH_ASM_T 7(asm_tac o list_∀_elim[⌜Front(Fst u)⌝,⌜cleanDirectory c1 y''⌝]));
a(LEMMA_T⌜(∃ z
            • (c1 dominates Dir_exist z ∧ (Front (Fst u), z) ∈ repState s1)
              ∧ cleanDirectory c1 y'' = cleanDirectory c1 z)⌝asm_tac);
(* *** Goal "3.2.3.2.1.1" *** *)
a(∃_tac⌜y''⌝THEN asm_rewrite_tac[]);
a(fc_tac[dominates_trans] THEN asm_fc_tac[]);
(* *** Goal "3.2.3.2.1.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(lemma_tac⌜z' = y⌝);
(* *** Goal "3.2.3.2.1.2.1" *** *)
a(DROP_NTH_ASM_T 33(asm_tac o rewrite_rule[functional_def]));
a(POP_ASM_T (strip_asm_tac o list_∀_elim[⌜Front (Fst u)⌝,⌜z'⌝,⌜y⌝]));
(*
```

SML

```
*)
(* *** Goal "3.2.3.2.1.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac);
a(⇒_T (strip_asm_tac o rewrite_rule[cleanDirectory_def,
        get_spec⌜MkDirectory⌝,dir_components]));
a(∃_tac⌜MkDirectory (Dir_tables y''
                          ⊕ {(Last (Fst u),
                               replaceRows
                                   y'''
                                   (RelList
                                       (ListRel (TS_rows y''')
                                           ⊕ (RelCombine
                                               ((revealRow c2 y''')~
                                                   ⨾ Snd u)
                                               (ListRel (TS_rows y'''))
                                             ⨾ Graph
                                               (updateRow
                                                   c2
                                                   (TS_class y')))
                                           ⨾ Graph destVal)))})
                    (Dir_exist y'')
                    (Dir_class y'')⌝THEN asm_rewrite_tac[get_spec⌜MkDirectory⌝]);
(*
```

SML

```
*)
a(lemma_tac⌜c1 dominates Dir_exist y⌝ THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(rewrite_tac[cleanDirectory_def]);
a(lemma_tac⌜c1 dominates Dir_class y''⌝ THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(asm_rewrite_tac[get_spec⌜MkDirectory⌝,dir_components]);
a(lemma_tac⌜c1 dominates Dir_class y⌝ THEN_LIST
        [fc_tac[dominates_trans] THEN asm_fc_tac[],asm_rewrite_tac[]]);
a(DROP_NTH_ASM_T 6 ante_tac THEN asm_rewrite_tac[rel_ext_clauses]THEN ⇒_tac);
a(rewrite_tac[⊕_single]);
a(REPEAT ∀_tac THEN ⇔_T asm_tac);
(*
```

SML

```
*)
(* *** Goal "3.2.3.2.1.2.2.1" *** *)
a(cases_tac⌜x' = Last (Fst u)⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.3.2.1.2.2.1.1" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜replaceRows y'''
                (RelList
                        (ListRel (TS_rows y''')
                            ⊕ (RelCombine
                                ((revealRow c2 y''')~ ⨾ Snd u)
                                (ListRel (TS_rows y'''))
                              ⨾ Graph (updateRow c2 (TS_class y')))
                            ⨾ Graph destVal))⌝ THEN asm_rewrite_tac[]);
a(ante_tac(list_∀_elim[⌜c1⌝,⌜c2⌝,⌜y'''⌝,⌜y'⌝,⌜Snd u⌝]cleanTable_updateRows_lemma)
        THEN asm_rewrite_tac[] THEN ⇒_T rewrite_thm_tac);
(* *** Goal "3.2.3.2.1.2.2.1.2" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜z''⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.3.2.1.2.2.2" *** *)
a(cases_tac⌜x' = Last (Fst u)⌝THEN asm_rewrite_tac[]);
(*
```

SML

```
*)
(* *** Goal "3.2.3.2.1.2.2.2.1" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(∃_tac⌜replaceRows y'
                    (RelList
                        (ListRel (TS_rows y')
                            ⊕ (RelCombine
                                ((revealRow c2 y')~ ⨾ Snd u)
                                (ListRel (TS_rows y'))
                              ⨾ Graph (updateRow c2 (TS_class y')))
                            ⨾ Graph destVal))⌝ THEN asm_rewrite_tac[]);
a(ante_tac(list_∀_elim[⌜c1⌝,⌜c2⌝,⌜y'''⌝,⌜y'⌝,⌜Snd u⌝]cleanTable_updateRows_lemma)
        THEN asm_rewrite_tac[]);
(* *** Goal "3.2.3.2.1.2.2.2.2" *** *)
a(DROP_NTH_ASM_T 2 strip_asm_tac);
a(DROP_NTH_ASM_T 4 (asm_tac o list_∀_elim[⌜x'⌝,⌜cleanTable c1 z''⌝]));
a(LEMMA_T⌜∃ z • (x', z) ∈ Dir_tables y''
                ∧ cleanTable c1 z'' = cleanTable c1 z⌝asm_tac);
(*
```

SML

```
*)
(* *** Goal "3.2.3.2.1.2.2.2.2.1" *** *)
a(∃_tac⌜z''⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.3.2.1.2.2.2.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(∃_tac⌜z'''⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.3.2.2" *** *)
a(DROP_NTH_ASM_T 3(strip_asm_tac o rewrite_rule
        [⊕_single,get_spec⌜MkDirectory⌝,dir_components]));
a(DROP_NTH_ASM_T 5(asm_tac o list_∀_elim[⌜x⌝,⌜cleanDirectory c1 z⌝]));
a(LEMMA_T⌜∃ z'
            • (c1 dominates Dir_exist z' ∧ (x, z') ∈ repState s2)
                ∧ cleanDirectory c1 z = cleanDirectory c1 z'⌝asm_tac);
(* *** Goal "3.2.3.2.2.1" *** *)
a(∃_tac⌜z⌝THEN asm_rewrite_tac[]);
(* *** Goal "3.2.3.2.2.2" *** *)
a(DROP_NTH_ASM_T 2 ante_tac THEN POP_ASM_T rewrite_thm_tac THEN ⇒_tac);
a(∃_tac⌜z'⌝THEN asm_rewrite_tac[]);
val conjunct2 = save_pop_thm"conjunct2";
```

HOL output

```
conjunct2 =
⊢ ∀ c1 c2 s1 s2 e
  • hideR (c1, repState s1) = hideR (c1, repState s2) ∧ c1 dominates c2
      ⇒ hideR (c1, Fst (updateStateR (c2, e, repState s1)))
        = hideR (c1, Fst (updateStateR (c2, e, repState s2)))
```

4 CLOSING DOWN

The following ProofPower instruction restores the previous proof context.

SML

```
pop_pc();
```

5 THE THEORY fef013

5.1 Parents

fef012

5.2 Children

fef015

5.3 Theorems

cleanTable_insertRows_lemma

$$
\begin{array}{l}
\vdash \forall\ c_1\ c_2\ t_1\ t_2\ ds \\
\quad \bullet\ cleanTable\ c_1\ t_1 = cleanTable\ c_1\ t_2 \\
\qquad \wedge\ c_1\ dominates\ c_2 \\
\qquad \Rightarrow cleanTable \\
\qquad\quad c_1 \\
\qquad\quad (replaceRows \\
\qquad\qquad t_1 \\
\qquad\qquad (TS\_rows\ t_1 \\
\qquad\qquad\quad @\ Map \\
\qquad\qquad\quad\ (MkRow\ c_2\ o\ colDefaults\ c_2\ t_1) \\
\qquad\qquad\quad\ ds)) \\
\qquad = cleanTable \\
\qquad\quad c_1 \\
\qquad\quad (replaceRows \\
\qquad\qquad t_2 \\
\qquad\qquad (TS\_rows\ t_2 \\
\qquad\qquad\quad @\ Map \\
\qquad\qquad\quad\ (MkRow\ c_2\ o\ colDefaults\ c_2\ t_2) \\
\qquad\qquad\quad\ ds))
\end{array}
$$

extract_⌢_single_lemma

$$
\begin{array}{l}
\vdash \forall\ c\ l\ x \\
\quad \bullet\ Extract \\
\qquad (1\ ..\ \#\ (l\ @\ [x]) \\
\qquad\quad \setminus\ Squash \\
\qquad\qquad (Id \\
\qquad\qquad\quad (Dom \\
\qquad\qquad\qquad (ListRel\ (l\ @\ [x]) \\
\qquad\qquad\qquad\quad \rhd \{r \\
\qquad\qquad\qquad\quad |c \\
\qquad\qquad\qquad\qquad dominates\ R\_exist \\
\qquad\qquad\qquad\qquad\ r\}))) \\
\qquad\qquad Image\ ns \\
\qquad\qquad \cap \{i|R\_exist\ (Nth\ (l\ @\ [x])\ i) = c\}) \\
\qquad (l\ @\ [x])
\end{array}
$$

```
= (if
    # l + 1
      ∈ 1 .. # (l @ [x])
        \ Squash
            (Id
                (Dom
                    (ListRel (l @ [x])
                        ▷ {r
                        |c
                            dominates R_exist
                              r})))
              Image ns
            ∩ {i|R_exist (Nth (l @ [x]) i) = c}
  then
    Extract
      (1 .. # l
          \ Squash
              (Id
                  (Dom
                      (ListRel l
                          ▷ {r
                          |c
                              dominates R_exist
                                r})))
            Image ns
          ∩ {i|R_exist (Nth l i) = c})
      l
    @ [x]
  else
    Extract
      (1 .. # l
          \ Squash
              (Id
                  (Dom
                      (ListRel l
                          ▷ {r
                          |c
                              dominates R_exist
                                r})))
            Image ns
          ∩ {i|R_exist (Nth l i) = c})
      l)
```

map_cleanRow_lemma1

```
⊢ ∀ l1 l2 c1 c2 s
  • c1 dominates c2
      ∧ Map
          (cleanRow c1 s)
          (l1 ↾ {r|c1 dominates R_exist r})
```

$$
\begin{array}{l}
\quad = Map \\
\quad\quad (cleanRow\ c_1\ s) \\
\quad\quad (l_2 \upharpoonright \{r | c_1\ dominates\ R\_exist\ r\}) \\
\Rightarrow Map \\
\quad\quad (cleanRow\ c_2\ s) \\
\quad\quad (l_1 \upharpoonright \{r | c_2\ dominates\ R\_exist\ r\}) \\
\quad = Map \\
\quad\quad (cleanRow\ c_2\ s) \\
\quad\quad (l_2 \upharpoonright \{r | c_2\ dominates\ R\_exist\ r\})
\end{array}
$$

map_cleanRow_lemma2

$$
\begin{array}{l}
\vdash \forall\ l_1\ l_2\ c_1\ c_2\ s \\
\quad \bullet\ c_1\ dominates\ c_2 \\
\quad\quad \wedge\ Map \\
\quad\quad\quad (cleanRow\ c_1\ s) \\
\quad\quad\quad (l_1 \upharpoonright \{r | c_1\ dominates\ R\_exist\ r\}) \\
\quad\quad = Map \\
\quad\quad\quad (cleanRow\ c_1\ s) \\
\quad\quad\quad (l_2 \upharpoonright \{r | c_1\ dominates\ R\_exist\ r\}) \\
\quad\quad \Rightarrow \#\ (ListRel\ l_1 \rhd \{r | c_2\ dominates\ R\_exist\ r\}) \\
\quad\quad\quad = \#\ (ListRel\ l_2 \rhd \{r | c_2\ dominates\ R\_exist\ r\})
\end{array}
$$

cleanTable_deleteRows_lemma

$$
\begin{array}{l}
\vdash \forall\ c_1\ c_2\ t_1\ t_2\ ns \\
\quad \bullet\ cleanTable\ c_1\ t_1 = cleanTable\ c_1\ t_2 \\
\quad\quad \wedge\ c_1\ dominates\ c_2 \\
\quad\quad \wedge\ c_2\ dominates\ TS\_class\ t_1 \\
\quad\quad \Rightarrow cleanTable \\
\quad\quad\quad c_1 \\
\quad\quad\quad (replaceRows \\
\quad\quad\quad\quad t_1 \\
\quad\quad\quad\quad (Extract \\
\quad\quad\quad\quad\quad (1\ ..\ \#\ (TS\_rows\ t_1) \\
\quad\quad\quad\quad\quad\quad \setminus revealRow\ c_2\ t_1\ Image\ ns \\
\quad\quad\quad\quad\quad\quad\quad \cap \{i \\
\quad\quad\quad\quad\quad\quad\quad | R\_exist\ (Nth\ (TS\_rows\ t_1)\ i) \\
\quad\quad\quad\quad\quad\quad\quad\quad = c_2\}) \\
\quad\quad\quad\quad\quad (TS\_rows\ t_1))) \\
\quad\quad = cleanTable \\
\quad\quad\quad c_1 \\
\quad\quad\quad (replaceRows \\
\quad\quad\quad\quad t_2 \\
\quad\quad\quad\quad (Extract \\
\quad\quad\quad\quad\quad (1\ ..\ \#\ (TS\_rows\ t_2) \\
\quad\quad\quad\quad\quad\quad \setminus revealRow\ c_2\ t_2\ Image\ ns \\
\quad\quad\quad\quad\quad\quad\quad \cap \{i \\
\quad\quad\quad\quad\quad\quad\quad | R\_exist\ (Nth\ (TS\_rows\ t_2)\ i) \\
\quad\quad\quad\quad\quad\quad\quad\quad = c_2\}) \\
\quad\quad\quad\quad\quad (TS\_rows\ t_2)))
\end{array}
$$

replaceData_updateField_lemma

```
⊢ ∀ c1 c2 d1 d2 t u
  • c1 dominates c2
      ∧ replaceData c1 d1 = replaceData c1 d2
      ∧ isVal (updateField c2 (TS_class t) (u, d1))
      ∧ isVal (updateField c2 (TS_class t) (u, d2))
    ⇒ replaceData
        c1
        (destVal
           (updateField c2 (TS_class t) (u, d1)))
      = replaceData
        c1
        (destVal
           (updateField c2 (TS_class t) (u, d2)))
```

cleanRow_updateRow_lemma1

```
⊢ ∀ c1 c2 r1 r2 t u
  • c1 dominates c2
      ∧ Dom u
        ⊆ {n
        |∃ c'
          • c' ∈ visibleCols c2 t ∧ CS_posn c' = n}
      ∧ cleanRow c1 (Snd (cleanColCons c1 t)) r1
        = cleanRow c1 (Snd (cleanColCons c1 t)) r2
      ∧ isVal (updateRow c2 (TS_class t) (u, r1))
      ∧ isVal (updateRow c2 (TS_class t) (u, r2))
    ⇒ cleanRow
        c1
        (Snd (cleanColCons c1 t))
        (destVal
           (updateRow c2 (TS_class t) (u, r1)))
      = cleanRow
        c1
        (Snd (cleanColCons c1 t))
        (destVal
           (updateRow c2 (TS_class t) (u, r2)))
```

inUpdates_lemma

```
⊢ ∀ l1 l2 x1 x2 c us up
  • R_exist x1 = R_exist x2
      ∧ # (ListRel l1 ▷ {r|c dominates R_exist r})
        = # (ListRel l2 ▷ {r|c dominates R_exist r})
    ⇒ ((#
          (Squash
             (Id
                (Dom
                   (ListRel (l1 @ [x1])
                      ▷ {r
                      |c
                          dominates R_exist
                          r}))))), up)
```

$$
\begin{array}{l}
\quad \in us \\
\Leftrightarrow (\# \\
\qquad (Squash \\
\qquad\quad (Id \\
\qquad\qquad (Dom \\
\qquad\qquad\quad (ListRel\ (l_2\ @\ [x_2]) \\
\qquad\qquad\qquad \rhd \{r \\
\qquad\qquad\qquad |c \\
\qquad\qquad\qquad\quad dominates\ R\_exist \\
\qquad\qquad\qquad\quad r\}))))),\ up) \\
\quad \in us)
\end{array}
$$

cleanTable_updateRows_lemma

$$
\begin{array}{l}
\vdash \forall\ c_1\ c_2\ t_1\ t_2\ us \\
\quad \bullet\ us \in Functional \\
\qquad \wedge\ Dom\ (\bigcup\ (Ran\ us)) \\
\qquad\quad \subseteq \{n \\
\qquad\quad |\exists\ c \\
\qquad\qquad \bullet\ c \in Snd\ (cleanColCons\ c_2\ t_2) \\
\qquad\qquad\quad \wedge\ CS\_posn\ c = n\} \\
\qquad \wedge\ ((RelCombine \\
\qquad\qquad\quad (revealRow\ c_2\ t_1\ ^{\sim}\ ⨾\ us) \\
\qquad\qquad\quad (ListRel\ (TS\_rows\ t_1)) \\
\qquad\qquad\ ⨾\ Graph\ (updateRow\ c_2\ (TS\_class\ t_2))) \\
\qquad\qquad \rhd \{x|isError\ x\}) \\
\qquad\quad ⨾\ Graph\ destError \\
\qquad\quad = \{\} \\
\qquad \wedge\ ((RelCombine \\
\qquad\qquad\quad (revealRow\ c_2\ t_2\ ^{\sim}\ ⨾\ us) \\
\qquad\qquad\quad (ListRel\ (TS\_rows\ t_2)) \\
\qquad\qquad\ ⨾\ Graph\ (updateRow\ c_2\ (TS\_class\ t_2))) \\
\qquad\qquad \rhd \{x|isError\ x\}) \\
\qquad\quad ⨾\ Graph\ destError \\
\qquad\quad = \{\} \\
\qquad \wedge\ cleanTable\ c_1\ t_1 = cleanTable\ c_1\ t_2 \\
\qquad \wedge\ c_1\ dominates\ c_2 \\
\qquad \wedge\ c_2\ dominates\ TS\_class\ t_1 \\
\quad \Rightarrow cleanTable \\
\qquad c_1 \\
\qquad (replaceRows \\
\qquad\quad t_1 \\
\qquad\quad (RelList \\
\qquad\qquad (ListRel\ (TS\_rows\ t_1) \\
\qquad\qquad\quad \oplus (RelCombine \\
\qquad\qquad\qquad (revealRow\ c_2\ t_1\ ^{\sim}\ ⨾\ us) \\
\qquad\qquad\qquad (ListRel\ (TS\_rows\ t_1)) \\
\qquad\qquad\qquad ⨾\ Graph \\
\qquad\qquad\qquad (updateRow \\
\qquad\qquad\qquad\quad c_2
\end{array}
$$

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad\qquad\qquad (TS\_class\ t_1)))\\
\qquad\qquad\qquad\qquad\qquad\ \fatsemi\ Graph\ destVal)))\\
\qquad\qquad = cleanTable\\
\qquad\qquad\quad c_1\\
\qquad\qquad\quad (replaceRows\\
\qquad\qquad\qquad\quad t_2\\
\qquad\qquad\qquad\quad (RelList\\
\qquad\qquad\qquad\qquad (ListRel\ (TS\_rows\ t_2)\\
\qquad\qquad\qquad\qquad\quad \oplus\ (RelCombine\\
\qquad\qquad\qquad\qquad\qquad\quad (revealRow\ c_2\ t_2\ \sim\ \fatsemi\ us)\\
\qquad\qquad\qquad\qquad\qquad\quad (ListRel\ (TS\_rows\ t_2))\\
\qquad\qquad\qquad\qquad\qquad\ \fatsemi\ Graph\\
\qquad\qquad\qquad\qquad\qquad\quad (updateRow\\
\qquad\qquad\qquad\qquad\qquad\qquad c_2\\
\qquad\qquad\qquad\qquad\qquad\qquad (TS\_class\ t_2)))\\
\qquad\qquad\qquad\qquad\qquad\ \fatsemi\ Graph\ destVal)))
\end{array}
$$

tabExists_lemma1

$$
\begin{array}{l}
\vdash \forall\ c_1\ c_2\ i\ s\\
\quad \bullet\ c_1\ dominates\ c_2 \wedge tabExists\ c_2\ i\ s\\
\qquad \Rightarrow tabExists\ c_1\ i\ s
\end{array}
$$

tabExists_cleanTable_lemma1

$$
\begin{array}{l}
\vdash \forall\ c\ s_1\ s_2\\
\quad \bullet\ hideR\ (c,\ repState\ s_1) = hideR\ (c,\ repState\ s_2)\\
\qquad \Rightarrow (\forall\ i\\
\qquad \bullet\ tabExists\ c\ i\ (repState\ s_1)\\
\qquad\quad \Rightarrow cleanTable\ c\ (getTable\ i\ (repState\ s_1))\\
\qquad\qquad = cleanTable\ c\ (getTable\ i\ (repState\ s_2)))
\end{array}
$$

colDefaults_lemma

$$
\begin{array}{l}
\vdash \forall\ c\ t_1\ t_2\\
\quad \bullet\ c\ dominates\ TS\_class\ t_1\\
\qquad\quad \wedge\ c\ dominates\ TS\_class\ t_2\\
\qquad\quad \wedge\ cleanTable\ c\ t_1 = cleanTable\ c\ t_2\\
\qquad \Rightarrow colDefaults\ c\ t_1 = colDefaults\ c\ t_2
\end{array}
$$

conjunct2

$$
\begin{array}{l}
\vdash \forall\ c_1\ c_2\ s_1\ s_2\ e\\
\quad \bullet\ hideR\ (c_1,\ repState\ s_1)\\
\qquad\quad = hideR\ (c_1,\ repState\ s_2)\\
\qquad\quad \wedge\ c_1\ dominates\ c_2\\
\qquad \Rightarrow hideR\\
\qquad\quad (c_1,\\
\qquad\qquad Fst\\
\qquad\qquad\quad (updateStateR\\
\qquad\qquad\qquad (c_2,\ e,\ repState\ s_1)))\\
\qquad\quad = hideR\\
\qquad\quad (c_1,\\
\qquad\qquad Fst\\
\qquad\qquad\quad (updateStateR\\
\qquad\qquad\qquad (c_2,\ e,\ repState\ s_2)))
\end{array}
$$

6 INDEX