```
POLYNOM_3
---------
*C polynom_3_begin            ************ POLYNOM_3 ************
*C omral_com   ==========================
               ORDERED-MONOID RING A-LISTS
               ==========================
               omral = o(rdered) m(onoid) a-l(ist)
*C omral_com_1 =======================
               DEFINITION OF OMRAL TYPE
               =======================
*D omralist_df                omral(<g:g:*>;<r:r:*>)== omralist{}(<g>; <r>)
*A omralist                   omral(g;r) ==  oal(g↓oset;r↓+gp)
*T omralist_wf  5.0 sec.
⊢ ∀g:OCMon. ∀r:CRng.  omral(g;r) ∈ DSet
|
BY (Unfold ‘omralist‘ 0 ...)
*M omralist_ml                note_reduction_strength ‘omralist‘ ‘8‘;;
*D omral_dom_df
               dom{<g:g:*>,<r:r:*>}(<ps:ps:*>)== omral_dom{}(<g>; <r>; <ps>)
               dom(<ps:ps:*>)== omral_dom{}(<g>; <r>; <ps>)
*A omral_dom                  dom(ps) ==  dom(ps)
*T omral_dom_wf  6.2 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps:(|g| × |r|) List.  dom(ps) ∈ MSet{g↓oset}
|
BY (Unfold ‘omral_dom‘ 0 ...)
*T omral_dom_wf2  8.8 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps:|omral(g;r)|.  dom(ps) ∈ FSet{g↓oset}
|
BY (Unfold ‘omral_dom‘ 0 ...)
*M omral_dom_eval
               let omral_dom_nilC =
                 MacroC ‘omral_dom_nilC‘
                 (EvalC ‘‘omral_dom‘‘)
                 ⌈dom([])⌉
                 IdC
                 ⌈0{g↓oset}⌉
               ;;
               let omral_dom_cons_prC =
                 MacroC ‘omral_cons_prC‘
                 (EvalC ‘‘omral_dom‘‘)
                 ⌈dom(<k, v>::ps)⌉
                 (UnfoldC ‘omral_dom‘)
                 ⌈mset_inj{g↓oset}(k) + dom(ps)⌉
               ;;
               add_AbReduce_conv ‘omral_dom‘
                 (omral_dom_nilC ORELSEC omral_dom_cons_prC)
               ;;
*T omralist_car_properties  5.7 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ws:|omral(g;r)|.  ↑sd_ordered(map(λx.x.1;ws)) ∧ ¬↑(0 ∈b map(λx.x.2;ws))
|
BY ProveSpecializedLemma ‘oalist_car_properties‘ 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] ((AbReduceIfC (\e t.not is_term ‘oalist‘ (subterm t 1))
     ANDTHENC TryC (Folds
   C
     ‘‘omralist omral_plus grp_lt grp_leq grp_blt‘‘)))
*M oal_to_omral
               % Lifting Theorems from oalists to omralists %
```

```
              let omral_opids =
              ``omralist omral_plus omral_dom grp_lt grp_leq grp_blt
                omral_zero omral_minus omral_inj``
              ;;
              let OmRalC =
               ForceReduceC `5` ANDTHENC TryC (FoldsC omral_opids) ;;
              let OmRalCStr =
                "ForceReduceC `5` ANDTHENC TryC (FoldsC ``"
                J
                concatenate_strings
                  (map (\id.tok_to_string id J " ") omral_opids)
                J
                "``)"
              ;;
              let mk_omral_thm old_name new_name new_pos =
               add_specialized_theorem
                old_name
                [`g',⌜OCMon⌝;`r',⌜CRng⌝]     % New outer context %
                [⌜parm{i}⌝;⌜g↓oset⌝;⌜r↓+gp⌝] % Bindings for outer context of old thm %
                OmRalC
                OmRalCStr
                new_name
                new_pos
                ; refresh()
              ;;
*T rng_before_imp_before_all  5.7 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀ps:|omral(g;r)|.
|    ↑before(k;map(λz.z.1;ps)) ⇒ ↑(∀_b x(:|g|) ∈ map(λz.z.1;ps). x <_b k)
|
BY ProveSpecializedLemma `before_imp_before_all` 2 [⌜parm{i}⌝;⌜g↓oset⌝;⌜r↓+gp⌝
   ] ((AbReduceIfC (\e t.not is_term `oalist` (subterm t 1))
     ANDTHENC FoldsC
     `
   `omralist omral_plus grp_lt grp_leq grp_blt``))
*T rng_before_all_imp_before  5.8 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀ps:(|g| × |r|) List.
|    ↑(∀_b x(:|g|) ∈ map(λz.z.1;ps). x <_b k) ⇒ ↑before(k;map(λz.z.1;ps))
|
BY ProveSpecializedLemma `before_all_imp_before` 2 [⌜parm{i}⌝;⌜g↓oset⌝;⌜r↓+gp⌝
   ] ((AbReduceIfC (\e t.not is_term `oalist` (subterm t 1))
     ANDTHENC FoldsC
     `
   `omralist omral_plus grp_lt grp_leq grp_blt``))
*T omralist_cases  5.5 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀Q:|omral(g;r)| → ℙ.
|    Q[[]]
|    ⇒ (∀ws:|omral(g;r)|. ∀x:|g|. ∀y:|r|.
|          ↑before(x;map(λx.x.1;ws)) ⇒ ¬(y = 0) ⇒ Q[<x, y>::ws])
|    ⇒ {∀ws:|omral(g;r)|. Q[ws]}
|
BY (D 0 THENM D 0 ...a)
|  THEN AssertLemma `oalist_cases_a` []
|  THEN (With ⌜g↓oset⌝ (D (-1)) THENM With ⌜r↓+gp⌝ (D (-1)) ...a)
|
1. g: OCMon
2. r: CRng
3. ∀Q:|oal(g↓oset;r↓+gp)| → ℙ
```

```
    Q[[]]
    ⇒ (∀ws:|oal(g↓oset;r↓+gp)|. ∀x:|(g↓oset)|. ∀y:|r↓+gp|.
          ↑before(x;map(λx.x.1;ws)) ⇒ ¬(y = e) ⇒ Q[<x, y>::ws])
    ⇒ {∀ws:|oal(g↓oset;r↓+gp)|. Q[ws]}
⊢ ∀Q:|omral(g;r)| → ℙ
|    Q[[]]
|    ⇒ (∀ws:|omral(g;r)|. ∀x:|g|. ∀y:|r|.
|          ↑before(x;map(λx.x.1;ws)) ⇒ ¬(y = 0) ⇒ Q[<x, y>::ws])
|    ⇒ {∀ws:|omral(g;r)|. Q[ws]}
|
BY % This is too fiddly. Need better way of doing this %
   AbReduceIf (\e t.not is_term `oalist` (subterm t 1)) (-1)
   THEN Unfold `omralist` 0 THEN Trivial
*T omralist_ind_a  7.0 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀Q:|omral(g;r)| → ℙ.
|    Q[[]]
|    ⇒ (∀ws:|omral(g;r)|
|          Q[ws] ⇒ (∀x:|g|. ∀y:|r|.  ↑before(x;map(λx.x.1;ws)) ⇒ ¬(y = 0) ⇒ Q[<x, y>::ws]))
|    ⇒ {∀ws:|omral(g;r)|. Q[ws]}
|
BY ProveSpecializedLemma `oalist_ind_a` 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] ((AbReduceIfC (\e t.not is_term `oalist` (subterm t 1))
     ANDTHENC TryC (Folds
   C
     ``omralist omral_plus grp_lt grp_leq grp_blt``)))
*T omral_lookups_same_a  6.3 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:|omral(g;r)|.  (∀u:|g|. ps[u] = qs[u]) ⇒ ps = qs
|
BY ProveSpecializedLemma `lookups_same_a` 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] ((AbReduceIfC (\e t.not is_term `oalist` (subterm t 1))
     ANDTHENC TryC (Folds
   C
     ``omralist omral_plus grp_lt grp_leq grp_blt``)))
*T rng_lookup_before_start  5.3 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀ps:|omral(g;r)|.  ↑before(k;map(λz.z.1;ps)) ⇒ ps[k] = 0
|
BY (D 0 THENM D 0
|   THENM AssertLemma `lookup_before_start` []
|   THENM With ⌈g↓oset⌉ (D (-1)) THENM With ⌈r↓+gp⌉ (D (-1)) ...a)
|
1. g: OCMon
2. r: CRng
3. ∀k:|(g↓oset)|. ∀ps:|oal(g↓oset;r↓+gp)|.  ↑before(k;map(λz.z.1;ps)) ⇒ ps[k] = e
⊢ ∀k:|g|. ∀ps:|omral(g;r)|.  ↑before(k;map(λz.z.1;ps)) ⇒ ps[k] = 0
|
BY AbReduceIf (\e t.not is_term `oalist` (subterm t 1)) 3
|  THEN Fold `omralist` 3
|
3. ∀k:|g|. ∀ps:|omral(g;r)|.  ↑before(k;map(λz.z.1;ps)) ⇒ ps[k] = 0
|
BY Trivial
*T lookup_omral_eq_zero  6.2 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀ps:|omral(g;r)|.  ¬↑(k ∈b dom(ps)) ⇒ ps[k] = 0
|
BY ProveSpecializedLemma `lookup_oal_eq_id` 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] ((AbReduceIfC (\e t.not is_term `oalist` (subterm t 1))
     ANDTHENC TryC (Folds
```

```
   C
     ‘‘omralist omral_plus omral_dom grp_lt grp_leq grp_blt‘‘)))
*C omral_plus_com
                ===================
                OMRAL PLUS FUNCTION
                ===================
                Lifting of oal merge function
*D omral_plus_df
                Parens ::Prec(inop)::
                  <ps:ps:L> ++<g:g:L>,<r:r:L> <qs:qs:L>
                  == omral_plus{}(<g>; <r>; <ps>; <qs>)
                Parens ::Prec(inop)::
                  <ps:ps:L> ++ <qs:qs:L>
                  == omral_plus{}(<g>; <r>; <ps>; <qs>)
*A omral_plus                 ps ++ qs ==  ps ++ qs
*T omral_plus_wf  9.4 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:(|g| × |r|) List.  ps ++ qs ∈ (|g| × |r|) List
|
BY (Unfold ‘omral_plus‘ 0 THEN RepD ...a)
|
1. g: OCMon
2. r: CRng
3. ps: (|g| × |r|) List
4. qs: (|g| × |r|) List
⊢ ps ++ qs ∈ (|g| × |r|) List
|
BY % MemCD picks second wf lemma which is not wanted here %
   (BLemma ‘oal_merge_wf‘ ...)
*T omral_plus_sd_ordered  6.2 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:(|g| × |r|) List.
|    ↑sd_ordered(map(λx.x.1;ps))
|    ⇒ ↑sd_ordered(map(λx.x.1;qs))
|    ⇒ ↑sd_ordered(map(λx.x.1;ps ++ qs))
|
BY ProveSpecializedLemma ‘oal_merge_sd_ordered‘ 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] ((AbReduceIfC (\e t.not is_term ‘oalist‘ (subterm t 1))
     ANDTHENC TryC (Folds
   C
     ‘‘omralist omral_plus grp_lt grp_leq grp_blt‘‘)))
*T omral_plus_non_zero_vals  7.2 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:(|g| × |r|) List.
|    ¬↑(0 ∈b map(λx.x.2;ps)) ⇒ ¬↑(0 ∈b map(λx.x.2;qs)) ⇒ ¬↑(0 ∈b map(λx.x.2;ps ++ qs))
|
BY ProveSpecializedLemma ‘oal_merge_non_id_vals‘ 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] ((AbReduceIfC (\e t.not is_term ‘oalist‘ (subterm t 1))
     ANDTHENC TryC (Folds
   C
     ‘‘omralist omral_plus grp_lt grp_leq grp_blt‘‘)))
*T omral_plus_wf2  5.6 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:|omral(g;r)|.  ps ++ qs ∈ |omral(g;r)|
|
BY ProveSpecializedLemma ‘oal_merge_wf2‘ 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] ((AbReduceIfC (\e t.not is_term ‘oalist‘ (subterm t 1))
     ANDTHENC TryC (Folds
   C
     ‘‘omralist omral_plus omral_dom grp_lt grp_leq grp_blt‘‘)))
*T omral_plus_dom  6.1 sec.
```

```
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:|omral(g;r)|.  ↑(dom(ps ++ qs) ⊆b dom(ps) ∪ dom(qs))
|
BY ProveSpecializedLemma ‘oal_dom_merge‘ 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] ((AbReduceIfC (\e t.not is_term ‘oalist‘ (subterm t 1))
     ANDTHENC TryC (Folds
   C
     ‘‘omralist omral_plus omral_dom grp_lt grp_leq grp_blt‘‘)))
*T lookup_omral_plus  7.0 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀ps,qs:|omral(g;r)|.  (ps ++ qs)[k] = ps[k] +r qs[k]
|
BY ProveSpecializedLemma ‘lookup_merge‘ 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] ((AbReduceIfC (\e t.not is_term ‘oalist‘ (subterm t 1))
     ANDTHENC TryC (Folds
   C
     ‘‘omralist omral_plus grp_lt grp_leq grp_blt‘‘)))
*T omral_plus_comm  6.3 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:|omral(g;r)|.  ps ++ qs = qs ++ ps
|
BY ProveSpecializedLemma ‘oal_merge_comm‘ 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] (ForceReduceC ‘5‘ ANDTHENC TryC (FoldsC ‘‘omralist omral_plus omral_dom grp_lt
    grp_leq grp_blt omral_zero omral_minus omral_inj ‘‘))
*T omral_plus_assoc  7.2 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs,rs:|omral(g;r)|.  ps ++ (qs ++ rs) = (ps ++ qs) ++ rs
|
BY ProveSpecializedLemma ‘oal_merge_assoc‘ 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] (ForceReduceC ‘5‘ ANDTHENC TryC (FoldsC ‘‘omralist omral_plus omral_dom grp_lt
    grp_leq grp_blt omral_zero omral_minus omral_inj ‘‘))
*C omral_zmi_com
               =========================================
               OMRAL ZERO, MINUS AND INJECTION FUNCTIONS
               =========================================
               All lifted from oal development.
*D omral_zero_df              00<g:g:*>,<r:r:*>== omral_zero{}(<g>; <r>)
*A omral_zero                 00g,r ==  00
*T omral_zero_wf  4.7 sec.
⊢ ∀g:OCMon. ∀r:CRng.  00g,r ∈ |omral(g;r)|
|
BY (Unfolds ‘‘omral_zero omralist‘‘ 0 ...)
*D omral_minus_df
               Parens ::Prec(preop)::
                 --<g:g:L>,<r:r:L> <ps:ps:L>
                 == omral_minus{}(<g>; <r>; <ps>)
               Parens ::Prec(preop)::  --<ps:ps:L>== omral_minus{}(<g>; <r>; <ps>)
*A omral_minus                --ps ==  --ps
*T omral_minus_wf  17.0 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps:|omral(g;r)|.  --ps ∈ |omral(g;r)|
|
BY (Unfold ‘omral_minus‘ 0 ...)
*D omral_inj_df
               inj{<g:g:*>,<r:r:*>}(<k:k:*>,<v:v:*>)== omral_inj{}(<g>; <r>; <k>; <v>)
               inj(<k:k:*>,<v:v:*>)== omral_inj{}(<g>; <r>; <k>; <v>)
*A omral_inj                  inj(k,v) ==  inj(k,v)
*T omral_inj_wf  10.7 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀v:|r|.  inj(k,v) ∈ |omral(g;r)|
|
BY (Unfold ‘omral_inj‘ 0 ...)
*T omral_dom_inj  6.8 sec.
```

```
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀v:|r|.
|    dom(inj(k,v)) = if v =b 0 then 0{g↓oset} else mset_inj{g↓oset}(k) fi
|
BY ProveSpecializedLemma ‘oal_dom_inj‘ 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] (ForceReduceC ‘5‘ ANDTHENC TryC (FoldsC ‘‘omralist omral_plus omral_dom grp_lt
    grp_leq grp_blt omral_zero omral_minus omral_inj ‘‘))
*T lookup_omral_inj  6.7 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k,k’:|g|. ∀v:|r|.  inj(k,v)[k’] = when k =b k’. v
|
BY ProveSpecializedLemma ‘lookup_oal_inj‘ 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] (ForceReduceC ‘5‘ ANDTHENC TryC (FoldsC ‘‘omralist omral_plus omral_dom grp_lt
    grp_leq grp_blt omral_zero omral_minus omral_inj ‘‘))
*T comb_for_omral_inj_wf  1.4 sec.
⊢ (λg,r,k,v,z.inj(k,v)) ∈ g:OCMon → r:CRng → k:|g| → v:|r| → ↓True → |omral(g;r)|
|
BY ProveOpCombTyping ‘omral_inj_wf‘
*T omral_fact  7.0 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps:|omral(g;r)|.
|    ps = msFor{oal_mon(g↓oset;r↓+gp)} k’ ∈ dom(ps). inj(k’,ps[k’])
|
BY ProveSpecializedLemma ‘oalist_fact‘ 2 [⌈parm{i}⌉;⌈g↓oset⌉;⌈r↓+gp⌉
   ] (ForceReduceC ‘5‘ ANDTHENC TryC (FoldsC ‘‘omralist omral_plus omral_dom grp_lt
    grp_leq grp_blt omral_zero omral_minus omral_inj ‘‘))
*T omral_fact_a  4.9 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps:|omral(g;r)|.
|    ps = msFor{omral_alg(g;r)↓grp} k’ ∈ dom(ps). inj(k’,ps[k’])
|
BY % Uggh ! %
|  RWH (MacroC ‘x‘
|    (EvalC ‘‘mset_for mon_for‘‘
|     ANDTHENC UnfoldsC ‘‘omral_plus omral_zero‘‘) ⌈msFor{omral_alg(g;r)↓grp} x ∈ a
|                                                      f[x]⌉
|     (EvalC ‘‘mset_for mon_for‘‘)
|    ⌈msFor{oal_mon(g↓oset;r↓+gp)} x ∈ a. f[x]⌉) 0
|
⊢ ∀g:OCMon. ∀r:CRng. ∀ps:|omral(g;r)|.
|    ps = msFor{oal_mon(g↓oset;r↓+gp)} k’ ∈ dom(ps). inj(k’,ps[k’])
|
BY Lemma ‘omral_fact‘
*C omral_scale_com
              ======================
              OMRAL SCALING FUNCTION
              ======================
              Scales keys and values of an omralist.
*D omral_scale_df
              Parens ::Prec(preop)::
                <<k:k:*>,<v:v:*>>*<g:mon:L>,<r:rng:L> <ps:ps:E>
                == omral_scale{}(<g>; <r>; <k>; <v>; <ps>)
              Parens ::Prec(preop)::
                <<k:k:*>,<v:v:*>>* <ps:ps:E>
                == omral_scale{}(<g>; <r>; <k>; <v>; <ps>)
*M omral_scale_ml
              <k,v>* ps
              ==r case ps of
                    [] => []
                    p::ps’ => if (v * p.2) =b 0
                              then <k,v>* ps’
```

```
                              else <k * p.1, v * p.2>::(<k,v>* ps')
                              fi
                  esac
*M omral_scale_eval
              let omral_scale_nilC =
                FwdMacroC `omral_scale_nilC`
                (RecEvalC ``omral_scale``) ⌈<k,v>* []⌉ ;;
              let omral_scale_cons_prC =
                FwdMacroC `omral_scale_cons_prC`
                (RecEvalC ``omral_scale``) ⌈<k,v>* (<k', v'>::ps)⌉ ;;
              add_AbReduce_conv `omral_scale`
                (omral_scale_nilC ORELSEC omral_scale_cons_prC) ;;
*T omral_scale_wf  4.2 sec.
⊢ ∀g:GrpSig. ∀r:RngSig. ∀k:|g|. ∀v:|r|. ∀ps:(|g| × |r|) List.  <k,v>* ps ∈ (|g| × |r|) List
|
BY (RepD
|   THENM New [`p';`ps\''] (OnVar `ps' ListInd)
|   THENM AbReduce 0 ...a)
|\
| 1. g: GrpSig
| 2. r: RngSig
| 3. k: |g|
| 4. v: |r|
| 5. ps: (|g| × |r|) List
| ⊢ [] ∈ (|g| × |r|) List
| |
1 BY Auto
 \
  1. g: GrpSig
  2. r: RngSig
  3. k: |g|
  4. v: |r|
  5. ps: (|g| × |r|) List
  6. p: |g| × |r|
  7. ps': (|g| × |r|) List
  8. <k,v>* ps' ∈ (|g| × |r|) List
  ⊢ <k,v>* (p::ps') ∈ (|g| × |r|) List
  |
  BY % Have to be careful here. Exists a later wf lemma
  |  which would also apply. The price to pay for having
  |  no half-way decent library object dependency tracking. %
  |
  |
  |  (D 6 THENM AbReduce 0
  |   THENM MemCD   ...a)
  |\
  | 6. p1: |g|
  | 7. p2: |r|
  | 8. ps': (|g| × |r|) List
  | 9. <k,v>* ps' ∈ (|g| × |r|) List
  | ⊢ (v * p2) =_b 0 ∈ 𝔹
  | |
  1 BY Auto
  |\
  | 6. p1: |g|
  | 7. p2: |r|
  | 8. ps': (|g| × |r|) List
```

```
  | 9. <k,v>* ps' ∈ (|g| × |r|) List
  | ⊢ <k,v>* ps' ∈ (|g| × |r|) List
  | |
  1 BY Trivial
   \
    6. p1: |g|
    7. p2: |r|
    8. ps': (|g| × |r|) List
    9. <k,v>* ps' ∈ (|g| × |r|) List
    ⊢ (<k * p1, v * p2>::(<k,v>* ps')) ∈ (|g| × |r|) List
    |
    BY (MemCD ...a)
    |\
    | ⊢ <k * p1, v * p2> ∈ |g| × |r|
    | |
    1 BY Auto
     \
      ⊢ <k,v>* ps' ∈ (|g| × |r|) List
      |
      BY Auto
*T omral_scale_dom_pred  27.9 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀Q:|g| → 𝔹. ∀k:|g|. ∀v:|r|. ∀ps:(|g| × |r|) List.
|    ↑(∀_b x(:|g|) ∈ map(λz.z.1;ps). Q[k * x]) ⇒ ↑(∀_b x(:|g|) ∈ map(λz.z.1;<k,v>* ps). Q[x])
|
BY (CDToVarThen 'ps' ListIndA ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| 3. Q: |g| → 𝔹
| 4. k: |g|
| 5. v: |r|
| ⊢ ↑(∀_b x(:|g|) ∈ map(λz.z.1;[]). Q[k * x]) ⇒ ↑(∀_b x(:|g|) ∈ map(λz.z.1;<k,v>* []). Q[x])
| |
1 BY (Reduce 0 ...)
 \
  1. g: OCMon
  2. r: CRng
  3. Q: |g| → 𝔹
  4. k: |g|
  5. v: |r|
  6. p: |g| × |r|
  7. ps: (|g| × |r|) List
  8. ↑(∀_b x(:|g|) ∈ map(λz.z.1;ps). Q[k * x]) ⇒ ↑(∀_b x(:|g|) ∈ map(λz.z.1;<k,v>* ps). Q[x])
  ⊢ ↑(∀_b x(:|g|) ∈ map(λz.z.1;p::ps). Q[k * x])
  |  ⇒ ↑(∀_b x(:|g|) ∈ map(λz.z.1;<k,v>* (p::ps)). Q[x])
  |
  BY New ['kp';'vp'] (D 6) THEN Reduce 0
  |  THEN (D 0 THENM RW bool_to_propC (-1)
  |        THENM D (-1) ...a)
  |
  6. kp: |g|
  7. vp: |r|
  8. ps: (|g| × |r|) List
  9. ↑(∀_b x(:|g|) ∈ map(λz.z.1;ps). Q[k * x]) ⇒ ↑(∀_b x(:|g|) ∈ map(λz.z.1;<k,v>* ps). Q[x])
  10. ↑Q[k * kp]
  11. ↑(∀_b x(:|g|) ∈ map(λz.z.1;ps). Q[k * x])
  ⊢ ↑(∀_b x(:|g|) ∈ map(λz.z.1;if (v * vp) =_b 0
```

```
  |      then <k,v>* ps
  |      else <k * kp, v * vp>::(<k,v>* ps)
  |      fi )
  |       Q[x])
  |
  BY (SplitOnConclITE THENM Reduce 0 ...a)
  |\
  | 12. v * vp = 0
  | ⊢ ↑(∀_b x(:|g|) ∈ map(λz.z.1;<k,v>* ps). Q[x])
  | |
  1 BY (HypBackchain ...)
   \
    12. ¬(v * vp = 0)
    ⊢ ↑(Q[k * kp] ∧_b (∀_b x(:|g|) ∈ map(λz.z.1;<k,v>* ps). Q[x]))
    |
    BY (RW bool_to_propC 0
        THENM HypBackchain ...)
*T omral_dom_scale  93.5 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀v:|r|. ∀ps:|omral(g;r)|.
|    ↑(dom(<k,v>* ps) ⊆_b fs-map(λk'.k' * k, dom(ps)))
|
BY (RepD THENM BLemma `mem_bsubmset` THENM RepD ...a)
|
1. g: OCMon
2. r: CRng
3. k: |g|
4. v: |r|
5. ps: |omral(g;r)|
6. x: |(g↓oset)|
7. ↑(x ∈_b dom(<k,v>* ps))
⊢ ↑(x ∈_b fs-map(λk'.k' * k, dom(ps)))
|
BY (Negate 0 THENM Negate 7 ...a)
|
7. ¬↑(x ∈_b fs-map(λk'.k' * k, dom(ps)))
⊢ ¬↑(x ∈_b dom(<k,v>* ps))
|
BY (Unfold `fset_map` 7
|   THENM RWH (LemmaC `fset_of_mset_mem`) 7 ...a)
|
7. ¬↑(x ∈_b msmap{g↓oset,g↓oset}(λk'.k' * k;dom(ps)))
|
BY RepUnfolds ``omral_dom oal_dom`` 7
|  THENM (RWH (LemmaC `mset_map_char`) 7 ...a)
|
7. ¬↑(x ∈_b mk_mset(map(λk'.k' * k;map(λz.z.1;ps))))
|
BY (RWH (LemmaWithC [`C',⌈|g|⌉] `map_map`) 7 ...a)
|
7. ¬↑(x ∈_b mk_mset(map((λk'.k' * k) o (λz.z.1);ps)))
|
BY Unfold `compose` 7 THEN Reduce 7
|
7. ¬↑(x ∈_b mk_mset(map(λx.x.1 * k;ps)))
|
BY % blow away mset stuff %
|
```

```
|  Reduce 6 THEN RenameVar `k\'' 6
|  THEN OnCls [0;7] (RepUnfolds ``omral_dom oal_dom mk_mset mset_mem mem``)
|  THEN OnCls [0;7] (Fold `bexists`)
|
6. k': |g|
7. ¬↑(∃_b x(:|(g↓oset)|) ∈ map(λx.x.1 * k;ps). x =_b k')
⊢ ¬↑(∃_b x(:|(g↓oset)|) ∈ map(λz.z.1;<k,v>* ps). x =_b k')
|
BY (OnMCls [0;7] (RW (SweepDnC
|     (LemmaC `bnot_thru_exists`
|      ORELSEC RevLemmaC `assert_of_bnot`))) ...a)
|
7. ↑(∀_b x(:|(g↓oset)|) ∈ map(λx.x.1 * k;ps). ¬_b(x =_b k'))
⊢ ↑(∀_b x(:|(g↓oset)|) ∈ map(λz.z.1;<k,v>* ps). ¬_b(x =_b k'))
|
BY (OnCls [0;7] Reduce THENM BLemma `omral_scale_dom_pred` ...a)
|
7. ↑(∀_b x(:|g|) ∈ map(λx.x.1 * k;ps). ¬_b(x =_b k'))
⊢ ↑(∀_b x(:|g|) ∈ map(λz.z.1;ps). ¬_b((k * x) =_b k'))
|
BY % Push both map funs onto mon_for arg %
|  (OnMCls [0;7]
|   (\i.Unfold `ball` i
|   THENM RWH (LemmaC `mon_for_map`) i
|   THENM Fold `ball` i) ...a)
|
7. ↑(∀_b x(:|(g↓oset × r↓+gp↓set)|) ∈ ps. ¬_b(((λx.x.1 * k) x) =_b k'))
⊢ ↑(∀_b x(:|(g↓oset × r↓+gp↓set)|) ∈ ps. ¬_b((k * ((λz.z.1) x)) =_b k'))
|
BY OnCls [0;7] Reduce
|
7. ↑(∀_b x(:|g| × |r|) ∈ ps. ¬_b((x.1 * k) =_b k'))
⊢ ↑(∀_b x(:|g| × |r|) ∈ ps. ¬_b((k * x.1) =_b k'))
|
BY (RWH (LemmaC `abmonoid_comm`) 0 ...)
*T omral_scale_dom_bound  33.5 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀bound,k:|g|. ∀v:|r|. ∀ps:(|g| × |r|) List.
|     ↑(∀_b x(:|g|) ∈ map(λz.z.1;ps). x <_b bound)
|     ⇒ ↑(∀_b x(:|g|) ∈ map(λz.z.1;<k,v>* ps). x <_b k * bound)
|
BY (RepD THENM BLemma `omral_scale_dom_pred` ...a)
|
1. g: OCMon
2. r: CRng
3. bound: |g|
4. k: |g|
5. v: |r|
6. ps: (|g| × |r|) List
7. ↑(∀_b x(:|g|) ∈ map(λz.z.1;ps). x <_b bound)
⊢ ↑(∀_b x(:|g|) ∈ map(λz.z.1;ps). k * x <_b k * bound)
|
BY % This is so ugly! (ball_char wants to match a set_car,
|    not a grp_car) %
|  (Assert ⌈|g| = |(g↓oset)|⌉ THENA Reduce 0 ...)
|  THEN (OnMCls [0;7] (RewriteWith [-1] ``ball_char``) ...a)
|  THEN Thin (-1)
|
```

```
7. ∀x:|(g↓oset)|. ↑(x ∈b map(λz.z.1;ps)) ⇒ ↑(x <b bound)
⊢ ∀x:|(g↓oset)|. ↑(x ∈b map(λz.z.1;ps)) ⇒ ↑(k * x <b k * bound)
|
BY (RepD THENM RW bool_to_propC 0 ...a)
|
8. x: |(g↓oset)|
9. ↑(x ∈b map(λz.z.1;ps))
⊢ k * x < k * bound
|
BY (BLemma ‘grp_op_preserves_lt‘
    THENM RWH (RevLemmaC ‘assert_of_grp_blt‘) 0
    THENM BHyp 7 ...)
*C omral_scale_sd_ordered_com
               The proof here needs some cleaning up.
               Probably, worth pulling out the second
               induction and generalizing it a bit.
*T omral_scale_sd_ordered  109.9 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀v:|r|. ∀ps:(|g| × |r|) List.
|    ↑sd_ordered(map(λz.z.1;ps)) ⇒ ↑sd_ordered(map(λz.z.1;<k,v>* ps))
|
BY (CDToVarThen ‘ps’ (\i.New [‘q’;‘qs’] (ListInd i))
|    ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| 3. k: |g|
| 4. v: |r|
| 5. ps: (|g| × |r|) List
| ⊢ ↑sd_ordered(map(λz.z.1;[])) ⇒ ↑sd_ordered(map(λz.z.1;<k,v>* []))
| |
1 BY AbReduce 0
| |
| ⊢ True ⇒ True
| |
1 BY Auto
 \
  1. g: OCMon
  2. r: CRng
  3. k: |g|
  4. v: |r|
  5. ps: (|g| × |r|) List
  6. q: |g| × |r|
  7. qs: (|g| × |r|) List
  8. ↑sd_ordered(map(λz.z.1;qs)) ⇒ ↑sd_ordered(map(λz.z.1;<k,v>* qs))
  ⊢ ↑sd_ordered(map(λz.z.1;q::qs)) ⇒ ↑sd_ordered(map(λz.z.1;<k,v>* (q::qs)))
  |
  BY New [‘kq’;‘vq’] (OnVar ‘q’ D) THEN AbReduce 0
  |  THEN (D 0 THENM RW bool_to_propC (-1) THENM D (-1) ...a)
  |  THEN (SplitOnConclITE ...a)
  |\
  | 6. kq: |g|
  | 7. vq: |r|
  | 8. qs: (|g| × |r|) List
  | 9. ↑sd_ordered(map(λz.z.1;qs)) ⇒ ↑sd_ordered(map(λz.z.1;<k,v>* qs))
  | 10. ↑before(kq;map(λz.z.1;qs))
  | 11. ↑sd_ordered(map(λz.z.1;qs))
  | 12. v * vq = 0
```

```
| ⊢ ↑sd_ordered(map(λz.z.1;<k,v>* qs))
| |
1 BY (BHyp 9 ...)
 \
  6. kq: |g|
  7. vq: |r|
  8. qs: (|g| × |r|) List
  9. ↑sd_ordered(map(λz.z.1;qs)) ⇒ ↑sd_ordered(map(λz.z.1;<k,v>* qs))
  10. ↑before(kq;map(λz.z.1;qs))
  11. ↑sd_ordered(map(λz.z.1;qs))
  12. ¬(v * vq = 0)
  ⊢ ↑sd_ordered(map(λz.z.1;<k * kq, v * vq>::(<k,v>* qs)))
  |
  BY % Can't use AbReduce as middle step because it destroys pattern
  |    RevLemmaC to match againsst %
  |
  |  (RWH (LemmaC `sd_ordered_char`) 0
  |   THENM RWH map_cons_unrollC 0
  |   THENM RWH mon_htfor_consC 0
  |   THENM RWH (RevLemmaC `sd_ordered_char`) 0 ...a)
  |\
  | 13. w: |(g↓oset)|
  | ⊢ (λz.z.1) <k * kq, v * vq> ∈ |(g↓oset)|
  | |
  1 BY (AbReduce 0 ...)
   \
    ⊢ ↑((∀_b w(:|(g↓oset)|) ∈ map(λz.z.1;<k,v>* qs). w <_b (λz.z.1) <k * kq, v * vq>)
    |  * sd_ordered(map(λz.z.1;<k,v>* qs)))
    |
    BY AbReduce 0 THENM (RW bool_to_propC 0 THENM D 0 ...a)
    |\
    | ⊢ ↑(∀_b w(:|g|) ∈ map(λz.z.1;<k,v>* qs). w <_b k * kq)
    | |
    1 BY (Assert ⌜↑sd_ordered(map(λz.z.1;<kq, vq>::qs))⌝
    | |   THENA (AbReduce 0 THEN RW bool_to_propC 0 THENM Auto) ...a)
    | |  THEN (RWH (LemmaC `sd_ordered_char`) (-1) ...a)
    | |
    | 13. ↑(HTFor{<𝔹,∧_b>} v::vs ∈ map(λz.z.1;<kq, vq>::qs). ∀_b w(:|(g↓oset)|) ∈ vs. w <_b v)
    | |
    1 BY AbReduce (-1) THENM (RW bool_to_propC (-1) THENM RepD ...a)
    | |
    | 13. ↑(∀_b w(:|g|) ∈ map(λz.z.1;qs). w <_b kq)
    | 14. ↑(HTFor{<𝔹,∧_b>} v::vs ∈ map(λz.z.1;qs). ∀_b w(:|g|) ∈ vs. w <_b v)
    | |
    1 BY % The last few steps were pretty ugly. They should be cleaned up %
    | |  OnHyps [14;12;11;10;9] Thin
    | |  THEN (OnVar 'qs' (MoveDepHypsToConclFor ListInd)  ...a)
    | |\
    | | 9. ↑(∀_b w(:|g|) ∈ map(λz.z.1;[]). w <_b kq)
    | | ⊢ ↑(∀_b w(:|g|) ∈ map(λz.z.1;<k,v>* []). w <_b k * kq)
    | | |
    1 2 BY (AbReduce 0 ...)
    | |\
    | | ⊢ map(λz.z.1;[]) ∈ |g| List
    | | |
    1 2 BY % Sigh. The tribulations of implicit polymorphism
    | |      and a weak type inf routine %
```

```
| |    (AbReduce 0 ...)
|  \
|   9. u: |g| × |r|
|   10. v1: (|g| × |r|) List
|   11. ↑(∀bw(:|g|) ∈ map(λz.z.1;v1). w <b kq)
|       ⇒ {↑(∀bw(:|g|) ∈ map(λz.z.1;<k,v>* v1). w <b k * kq)}
|   12. ↑(∀bw(:|g|) ∈ map(λz.z.1;u::v1). w <b kq)
|   ⊢ ↑(∀bw(:|g|) ∈ map(λz.z.1;<k,v>* (u::v1)). w <b k * kq)
|   |
1   BY D 9 THEN AbReduce 0 THEN (SplitOnConclITE ...a)
|   |\
|   | 9. u1: |g|
|   | 10. u2: |r|
|   | 11. v1: (|g| × |r|) List
|   | 12. ↑(∀bw(:|g|) ∈ map(λz.z.1;v1). w <b kq)
|   |     ⇒ {↑(∀bw(:|g|) ∈ map(λz.z.1;<k,v>* v1). w <b k * kq)}
|   | 13. ↑(∀bw(:|g|) ∈ map(λz.z.1;<u1, u2>::v1). w <b kq)
|   | 14. v * u2 = 0
|   | ⊢ ↑(∀bw(:|g|) ∈ map(λz.z.1;<k,v>* v1). w <b k * kq)
|   | |
1   2 BY (BHyp 12 ...)
|   | |
|   | ⊢ ↑(∀bw(:|g|) ∈ map(λz.z.1;v1). w <b kq)
|   | |
1   2 BY (AbReduce 13 THENM RW bool_to_propC 13 ...)
|    \
|     9. u1: |g|
|     10. u2: |r|
|     11. v1: (|g| × |r|) List
|     12. ↑(∀bw(:|g|) ∈ map(λz.z.1;v1). w <b kq)
|         ⇒ {↑(∀bw(:|g|) ∈ map(λz.z.1;<k,v>* v1). w <b k * kq)}
|     13. ↑(∀bw(:|g|) ∈ map(λz.z.1;<u1, u2>::v1). w <b kq)
|     14. ¬(v * u2 = 0)
|     ⊢ ↑(∀bw(:|g|) ∈ map(λz.z.1;<k * u1, v * u2>::(<k,v>* v1)). w <b k * kq)
|     |
1     BY (AbReduce 0 THENM RW bool_to_propC 0
|     |   THENM D 0 ...a)
|     |\
|     | ⊢ k * u1 <g↓oset k * kq
|     | |
1     2 BY (AbReduce 13 THENM RW bool_to_propC 13 THENM RepD ...a)
|     | |
|     | 13. u1 <g↓oset kq
|     | 14. ↑(∀bw(:|g|) ∈ map(λz.z.1;v1). w <b kq)
|     | 15. ¬(v * u2 = 0)
|     | |
1     2 BY OnCls [0;13] (Fold `grp_lt`)
|     |    THEN (BLemma `grp_op_preserves_lt` ...)
|      \
|       ⊢ ↑(∀bw(:|g|) ∈ map(λz.z.1;<k,v>* v1). w <b k * kq)
|       |
1       BY (BHyp 12 ...)
|       |
|       ⊢ ↑(∀bw(:|g|) ∈ map(λz.z.1;v1). w <b kq)
|       |
1       BY (AbReduce 13 THENM RW bool_to_propC 13 ...)
 \
```

```
        ⊢ ↑sd_ordered(map(λz.z.1;<k,v>* qs))
        |
        BY (BHyp 9 ...)
*T omral_scale_non_zero_vals  43.7 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀v:|r|. ∀ps:(|g| × |r|) List.
|    ¬↑(0 ∈b map(λx.x.2;ps)) ⇒ ¬↑(0 ∈b map(λx.x.2;<k,v>* ps))
|
BY (CDToVarThen ‘ps’ (\i.New [‘q’;‘qs’] (ListInd i))
|    ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| 3. k: |g|
| 4. v: |r|
| 5. ps: (|g| × |r|) List
| ⊢ ¬↑(0 ∈b map(λx.x.2;[])) ⇒ ¬↑(0 ∈b map(λx.x.2;<k,v>* []))
| |
1 BY (AbReduce 0 ...)
 \
  1. g: OCMon
  2. r: CRng
  3. k: |g|
  4. v: |r|
  5. ps: (|g| × |r|) List
  6. q: |g| × |r|
  7. qs: (|g| × |r|) List
  8. ¬↑(0 ∈b map(λx.x.2;qs)) ⇒ ¬↑(0 ∈b map(λx.x.2;<k,v>* qs))
  ⊢ ¬↑(0 ∈b map(λx.x.2;q::qs)) ⇒ ¬↑(0 ∈b map(λx.x.2;<k,v>* (q::qs)))
  |
  BY (New [‘kq’;‘vq’] (OnVar ‘q’ D)
  |   THENM AbReduce 0 THENM RW bool_to_propC 0
  |   THENM D 0 THENM RWH (LemmaC ‘not_over_or‘) (-1) ...a)
  |
  6. kq: |g|
  7. vq: |r|
  8. qs: (|g| × |r|) List
  9. ¬↑(0 ∈b map(λx.x.2;qs)) ⇒ ¬↑(0 ∈b map(λx.x.2;<k,v>* qs))
  10. ¬(vq = 0) ∧ ¬↑(0 ∈b map(λx.x.2;qs))
  ⊢ ¬↑(0 ∈b map(λx.x.2;if (v * vq) =b 0 then <k,v>* qs else <k * kq, v * vq>::(<k,v>* qs) fi ))
  |
  BY (SplitOnConclITE THENM AbReduce 0 ...a)
  |\
  | 11. v * vq = 0
  | ⊢ ¬↑(0 ∈b map(λx.x.2;<k,v>* qs))
  | |
  1 BY (BHyp 9 ...)
   \
    11. ¬(v * vq = 0)
    ⊢ ¬↑(((v * vq) =b 0) ∨b(0 ∈b map(λx.x.2;<k,v>* qs)))
    |
    BY (RW bool_to_propC 0
        THENM ProveProp ...)
*T omral_scale_wf2  22.0 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀v:|r|. ∀ps:|omral(g;r)|.  <k,v>* ps ∈ |omral(g;r)|
|
BY (RepD ...a)
|
```

```
1. g: OCMon
2. r: CRng
3. k: |g|
4. v: |r|
5. ps: |omral(g;r)|
⊢ <k,v>* ps ∈ |omral(g;r)|
|
BY (OnCls [0;5] (Unfold ‘omralist‘)
|   THENM AddCarProperties 5
|   THENM AbReduce 0 THENM MemTypeCD ...a)
|\
| 5. ps: |oal(g↓oset;r↓+gp)|
| 6. ↑sd_ordered(map(λx.x.1;ps)) ∧ ¬↑(e ∈b map(λx.x.2;ps))
| ⊢ <k,v>* ps ∈ (|g| × |r|) List
| |
1 BY Auto
|\
| 5. ps: |oal(g↓oset;r↓+gp)|
| 6. ↑sd_ordered(map(λx.x.1;ps))
| 7. ¬↑(e ∈b map(λx.x.2;ps))
| ⊢ ↑sd_ordered(map(λx.x.1;<k,v>* ps))
| |
1 BY (BLemma ‘omral_scale_sd_ordered‘ ...)
 \
  5. ps: |oal(g↓oset;r↓+gp)|
  6. ↑sd_ordered(map(λx.x.1;ps))
  7. ¬↑(e ∈b map(λx.x.2;ps))
  ⊢ ¬↑(0 ∈b map(λx.x.2;<k,v>* ps))
  |
  BY AbReduce 7 THEN (BLemma ‘omral_scale_non_zero_vals‘ ...)
*T lookup_omral_scale_a  89.5 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k,k’:|g|. ∀v:|r|. ∀ps:|omral(g;r)|.  (<k,v>* ps)[k * k’] = v * ps[k’]
|
BY (RepeatMFor 5 (D 0) ...a)
|
1. g: OCMon
2. r: CRng
3. k: |g|
4. k’: |g|
5. v: |r|
⊢ ∀ps:|omral(g;r)|. (<k,v>* ps)[k * k’] = v * ps[k’]
|
BY (Unfold ‘omralist‘ 0
|   THEN BLemma ‘oalist_ind‘ ...a)
|\
| ⊢ (<k,v>* [])[k * k’] = v * [][k’]
| |
1 BY AbReduce 0
| |
| ⊢ 0 = v * 0
| |
1 BY (RW RngNormC 0 ...)
 \
  ⊢ ∀ps:|oal(g↓oset;r↓+gp)|
  |    (<k,v>* ps)[k * k’] = v * ps[k’]
  |    ⇒ (∀x:|(g↓oset)|. ∀y:|r↓+gp|.
  |          ↑before(x;map(λx.x.1;ps))
```

```
|           ⇒ ¬(y = e)
|           ⇒ (<k,v>* (<x, y>::ps))[k * k’] = v * (<x, y>::ps)[k’])
|
BY AbReduceIf (\e t.not is_term ‘oalist‘ (subterm t 1)) 0
|  THEN Fold ‘omralist‘ 0 THEN RenameBVars [‘x’,‘kp’;‘y’,‘kv’] 0
|  THEN (RepD ...a)
|
6. ps: |omral(g;r)|
7. (<k,v>* ps)[k * k’] = v * ps[k’]
8. kp: |g|
9. kv: |r|
10. ↑before(kp;map(λkp.kp.1;ps))
11. ¬(kv = 0)
⊢ if (v * kv) =b 0 then <k,v>* ps else <k * kp, v * kv>::(<k,v>* ps) fi [k * k’]
|  = v * if kp =b k’ then kv else ps[k’] fi
|
BY (SplitOnConclITEs ...a)
|\
| 12. v * kv = 0
| 13. kp = k’
| ⊢ (<k,v>* ps)[k * k’] = v * kv
| |
1 BY (RewriteWith [12;13] [‘rng_lookup_before_start‘] 0 ...a)
| |\
| | ⊢ ↑before(k * k’;map(λz.z.1;<k,v>* ps))
| | |
1 2 BY (BLemma ‘rng_before_all_imp_before‘
| | |    THENM BLemma ‘omral_scale_dom_bound‘
| | |    THENM BLemma ‘rng_before_imp_before_all‘ ...a)
| | |
| | ⊢ ↑before(k’;map(λz.z.1;ps))
| | |
1 2 BY (RWH (RevHypC 13) 0 ...)
|   \
|    ⊢ 0 = 0
|    |
1    BY Auto
|\
| 12. v * kv = 0
| 13. ¬(kp = k’)
| ⊢ (<k,v>* ps)[k * k’] = v * ps[k’]
| |
1 BY Trivial
|\
| 12. ¬(v * kv = 0)
| 13. kp = k’
| ⊢ (<k * kp, v * kv>::(<k,v>* ps))[k * k’] = v * kv
| |
1 BY (Reduce 0 THEN SplitOnConclITE ...a)
| |\
| | 14. k * kp = k * k’
| | ⊢ v * kv = v * kv
| | |
1 2 BY Auto
|   \
|    14. ¬(k * kp = k * k’)
|    ⊢ (<k,v>* ps)[k * k’] = v * kv
```

```
  |   |
  1   BY (D 14 THENM RWH (HypC 13) 0 ...)
   \
    12. ¬(v * kv = 0)
    13. ¬(kp = k')
    ⊢ (<k * kp, v * kv>::(<k,v>* ps))[k * k'] = v * ps[k']
    |
    BY (Reduce 0 THEN SplitOnConclITE ...a)
    |\
    | 14. k * kp = k * k'
    | ⊢ v * kv = v * ps[k']
    | |
    1 BY % Contradiction %
    |    (FLemma `ocmon_cancel` [14] ...)
     \
      14. ¬(k * kp = k * k')
      ⊢ (<k,v>* ps)[k * k'] = v * ps[k']
      |
      BY Trivial
*T lookup_omral_scale_b  98.9 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k,k':|g|. ∀v:|r|. ∀ps:(|g| × |r|) List.
|    ¬(∃d:|g|. ↑(d ∈_b dom(ps)) ∧ k * d = k') ⇒ (<k,v>* ps)[k'] = 0
|
BY (CDToVarThen `ps' ListIndA
|   THENM Reduce 0  ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| 3. k: |g|
| 4. k': |g|
| 5. v: |r|
| ⊢ ¬(∃d:|g|. False ∧ k * d = k') ⇒ 0 = 0
| |
1 BY Auto
 \
  1. g: OCMon
  2. r: CRng
  3. k: |g|
  4. k': |g|
  5. v: |r|
  6. p: |g| × |r|
  7. ps: (|g| × |r|) List
  8. ¬(∃d:|g|. ↑(d ∈_b dom(ps)) ∧ k * d = k') ⇒ (<k,v>* ps)[k'] = 0
  ⊢ ¬(∃d:|g|. ↑(d ∈_b dom(p::ps)) ∧ k * d = k') ⇒ (<k,v>* (p::ps))[k'] = 0
  |
  BY New [`kp';`vp'] (D 6) THEN Reduce 0
  |  THEN (D 0 THENM SplitOnConclITE ...a)
  |\
  | 6. kp: |g|
  | 7. vp: |r|
  | 8. ps: (|g| × |r|) List
  | 9. ¬(∃d:|g|. ↑(d ∈_b dom(ps)) ∧ k * d = k') ⇒ (<k,v>* ps)[k'] = 0
  | 10. ¬(∃d:|g|. ↑((kp =_b d) ∨_b(d ∈_b dom(ps))) ∧ k * d = k')
  | 11. v * vp = 0
  | ⊢ (<k,v>* ps)[k'] = 0
  | |
  1 BY (BHyp 9 THENM D 0 THENM D 10 THENM ExRepD ...a)
```

```
  | |
  | 10. v * vp = 0
  | 11. d: |g|
  | 12. ↑(d ∈b dom(ps))
  | 13. k * d = k'
  | ⊢ ∃d:|g|. ↑((kp =b d) ∨b(d ∈b dom(ps))) ∧ k * d = k'
  | |
  1 BY (With ⌈d⌉ (D 0) ...)
  | |
  | ⊢ ↑((kp =b d) ∨b(d ∈b dom(ps)))
  | |
  1 BY (RW bool_to_propC 0 THENM Sel 2 (D 0) ...)
   \
    6. kp: |g|
    7. vp: |r|
    8. ps: (|g| × |r|) List
    9. ¬(∃d:|g|. ↑(d ∈b dom(ps)) ∧ k * d = k') ⇒ (<k,v>* ps)[k'] = 0
    10. ¬(∃d:|g|. ↑((kp =b d) ∨b(d ∈b dom(ps))) ∧ k * d = k')
    11. ¬(v * vp = 0)
    ⊢ (<k * kp, v * vp>::(<k,v>* ps))[k'] = 0
    |
    BY (Reduce 0 THEN SplitOnConclITE ...a)
    |\
    | 12. k * kp = k'
    | ⊢ v * vp = 0
    | |
    1 BY % hyp 12 contradicts hyp 10 %
    | |  (D 10 THENM With ⌈kp⌉ (D 0) ...)
    | |
    | 10. ¬(v * vp = 0)
    | 11. k * kp = k'
    | ⊢ ↑((kp =b kp) ∨b(kp ∈b dom(ps)))
    | |
    1 BY (RW bool_to_propC 0 THENM Sel 1 (D 0) ...)
     \
      12. ¬(k * kp = k')
      ⊢ (<k,v>* ps)[k'] = 0
      |
      BY (BHyp 9 THENM D 0 THENM D 10 THENM ExRepD ...)
      |
      10. ¬(v * vp = 0)
      11. ¬(k * kp = k')
      12. d: |g|
      13. ↑(d ∈b dom(ps))
      14. k * d = k'
      ⊢ ∃d:|g|. ↑((kp =b d) ∨b(d ∈b dom(ps))) ∧ k * d = k'
      |
      BY (With ⌈d⌉ (D 0) ...)
      |
      ⊢ ↑((kp =b d) ∨b(d ∈b dom(ps)))
      |
      BY (RW bool_to_propC 0 THENM Sel 2 (D 0) ...)
*T lookup_omral_scale_c  75.7 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀z,k:|g|. ∀v:|r|. ∀ps:|omral(g;r)|.
|     (<k,v>* ps)[z] = msFor{r↓+gp} y ∈ dom(ps). when (k * y) =b z. v * ps[y]
|
BY (RepD ...a)
```

```
|
1. g: OCMon
2. r: CRng
3. z: |g|
4. k: |g|
5. v: |r|
6. ps: |omral(g;r)|
⊢ (<k,v>* ps)[z] = msFor{r↓+gp} y ∈ dom(ps). when (k * y) =_b z. v * ps[y]
|
BY % This predicate is constructively decidable, but no need to prove so,
|    since here there is no constructive content. %
|  (AddXM 1 THENM
|   Decide ⌜∃d:|g|. ↑(d ∈_b dom(ps)) ∧ k * d = z⌝ ...a)
|\
| 1. XM{i'}
| 2. g: OCMon
| 3. r: CRng
| 4. z: |g|
| 5. k: |g|
| 6. v: |r|
| 7. ps: |omral(g;r)|
| 8. ∃d:|g|. ↑(d ∈_b dom(ps)) ∧ k * d = z
| |
1 BY ExRepD
| |
| 8. d: |g|
| 9. ↑(d ∈_b dom(ps))
| 10. k * d = z
| |
1 BY Unfold `rng_when` 0 THEN
| |  (RW (SweepUpC
| |   (RevHypC 10
| |    ORELSEC LemmaC `lookup_omral_scale_a`
| |    ORELSEC LemmaWithC [`u',⌜d⌝] `fset_for_when_unique`)) 0 ...a)
| |\
| | ⊢ ↑((k * d) =_b (k * d))
| | |
1 2 BY (RW bool_to_propC 0 ...)
| |\
| | 11. v1: |(g↓oset)|
| | 12. ↑((k * v1) =_b (k * d))
| | 13. ↑(v1 ∈_b dom(ps))
| | ⊢ v1 = d
| | |
1 2 BY (RW bool_to_propC 12
| |     THENM FLemma `ocmon_cancel` [12]
| |     THENM Reduce 0  ...)
|  \
|   ⊢ v * ps[d] = v * ps[d]
|   |
1   BY Auto
 \
  1. XM{i'}
  2. g: OCMon
  3. r: CRng
  4. z: |g|
  5. k: |g|
```

```
  6. v: |r|
  7. ps: |omral(g;r)|
  8. ¬(∃d:|g|. ↑(d ∈b dom(ps)) ∧ k * d = z)
  |
  BY (Unfold ‘rng_when‘ 0 THEN
  |   RewriteWith [] ‘‘lookup_omral_scale_b mset_for_when_none‘‘ 0 ...a)
  |\
  | 9. x: |(g↓oset)|
  | 10. ↑(x ∈b dom(ps))
  | ⊢ ¬↑((k * x) =b z)
  | |
  1 BY (D 0 THENM RW bool_to_propC (-1)
  |     THENM D 8 THENM With ⌈x⌉ (D 0) ...)
   \
    ⊢ 0 = e
    |
    BY (Reduce 0 ...)
*T lookup_omral_scale_d  0.2 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀z,k:|g|. ∀v:|r|. ∀ps:|omral(g;r)|.
|    (<k,v>* ps)[z] = (Σy ∈ dom(ps). when (k * y) =b z. v * ps[y])
|
BY Unfold ‘rng_mssum‘ 0
|  THENM AssertLemma ‘lookup_omral_scale_c‘ []
|
1. ∀g:OCMon. ∀r:CRng. ∀z,k:|g|. ∀v:|r|. ∀ps:|omral(g;r)|.
     (<k,v>* ps)[z] = msFor{r↓+gp} y ∈ dom(ps). when (k * y) =b z. v * ps[y]
⊢ ∀g:OCMon. ∀r:CRng. ∀z,k:|g|. ∀v:|r|. ∀ps:|omral(g;r)|.
|    (<k,v>* ps)[z] = msFor{r↓+gp} y ∈ dom(ps). when (k * y) =b z. v * ps[y]
|
BY Unfold ‘oset_of_ocmon‘ 1 THEN Trivial
*C omral_times_com
               ====================
               OMRAL TIMES FUNCTION
               ====================
*D omral_times_df
               Parens ::Prec(inop)::
                 <ps:ps:L> **<g:g:L>,<r:r:L> <qs:qs:L>
                 == omral_times{}(<g>; <r>; <ps>; <qs>)
               Parens ::Prec(inop)::
                 <ps:ps:L> ** <qs:qs:L>
                 == omral_times{}(<g>; <r>; <ps>; <qs>)
*M omral_times_ml
               ps ** qs==r case ps of [] => [] | p::ps’ => <p.1,p.2>* qs ++ (ps’ ** qs) esac
*M omral_times_eval
               let omral_times_nilC =
                 FwdMacroC ‘omral_times_nilC‘
                 (RecEvalC ‘‘omral_times‘‘) ⌈[] ** qs⌉ ;;
               let omral_times_cons_prC =
                 FwdMacroC ‘omral_times_cons_prC‘
                 (RecEvalC ‘‘omral_times‘‘) ⌈(<k, v>::ps) ** qs⌉
               ;;
               add_AbReduce_conv ‘omral_times‘
                 (omral_times_nilC ORELSEC omral_times_cons_prC)
               ;;
*T omral_times_wf  4.2 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:(|g| × |r|) List.  ps ** qs ∈ (|g| × |r|) List
|
```

```
BY (RepD THENM New [‘p’;‘ps\’’] (OnVar ‘ps’ ListInd)
|   THEN AbReduce 0
|      ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| 3. ps: (|g| × |r|) List
| 4. qs: (|g| × |r|) List
| ⊢ [] ∈ (|g| × |r|) List
| |
1 BY Auto
 \
  1. g: OCMon
  2. r: CRng
  3. ps: (|g| × |r|) List
  4. qs: (|g| × |r|) List
  5. p: |g| × |r|
  6. ps’: (|g| × |r|) List
  7. ps’ ** qs ∈ (|g| × |r|) List
  ⊢ (p::ps’) ** qs ∈ (|g| × |r|) List
  |
  BY New [‘k’;‘v’] (D 5) THEN AbReduce 0
  |
  5. k: |g|
  6. v: |r|
  7. ps’: (|g| × |r|) List
  8. ps’ ** qs ∈ (|g| × |r|) List
  ⊢ <k,v>* qs ++ (ps’ ** qs) ∈ (|g| × |r|) List
  |
  BY MemCD THEN Auto
*T omral_times_sd_ordered  10.7 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:(|g| × |r|) List.
|    ↑sd_ordered(map(λz.z.1;qs)) ⇒ ↑sd_ordered(map(λz.z.1;ps ** qs))
|
BY (RepD THENM OnVar ‘ps’ ListIndA ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| 3. qs: (|g| × |r|) List
| 4. ↑sd_ordered(map(λz.z.1;qs))
| ⊢ ↑sd_ordered(map(λz.z.1;[] ** qs))
| |
1 BY (Reduce 0 ...)
 \
  1. g: OCMon
  2. r: CRng
  3. qs: (|g| × |r|) List
  4. ↑sd_ordered(map(λz.z.1;qs))
  5. p: |g| × |r|
  6. ps: (|g| × |r|) List
  7. ↑sd_ordered(map(λz.z.1;ps ** qs))
  ⊢ ↑sd_ordered(map(λz.z.1;(p::ps) ** qs))
  |
  BY New [‘kp’;‘vp’] (OnVar ‘p’ D)
  |  THEN Reduce 0 THEN (RepD ...a)
  |
  5. kp: |g|
```

```
  6. vp: |r|
  7. ps: (|g| × |r|) List
  8. ↑sd_ordered(map(λz.z.1;ps ** qs))
  ⊢ ↑sd_ordered(map(λz.z.1;<kp,vp>* qs ++ (ps ** qs)))
  |
  BY (Backchain ‘‘omral_plus_sd_ordered omral_scale_sd_ordered‘‘ ...)
*T omral_times_non_zero_vals  11.2 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:(|g| × |r|) List.
|    ¬↑(0 ∈b map(λx.x.2;qs)) ⇒ ¬↑(0 ∈b map(λx.x.2;ps ** qs))
|
BY (RepD THENM OnVar ‘ps’ ListIndA ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| 3. qs: (|g| × |r|) List
| 4. ¬↑(0 ∈b map(λx.x.2;qs))
| ⊢ ¬↑(0 ∈b map(λx.x.2;[] ** qs))
| |
1 BY (Reduce 0 THEN D 0 ...)
 \
  1. g: OCMon
  2. r: CRng
  3. qs: (|g| × |r|) List
  4. ¬↑(0 ∈b map(λx.x.2;qs))
  5. p: |g| × |r|
  6. ps: (|g| × |r|) List
  7. ¬↑(0 ∈b map(λx.x.2;ps ** qs))
  ⊢ ¬↑(0 ∈b map(λx.x.2;(p::ps) ** qs))
  |
  BY New [‘kp’;‘vp’] (OnVar ‘p’ D)
  |  THEN Reduce 0
  |
  5. kp: |g|
  6. vp: |r|
  7. ps: (|g| × |r|) List
  8. ¬↑(0 ∈b map(λx.x.2;ps ** qs))
  ⊢ ¬↑(0 ∈b map(λx.x.2;<kp,vp>* qs ++ (ps ** qs)))
  |
  BY (Backchain ‘‘
        omral_scale_non_zero_vals
        omral_plus_non_zero_vals‘‘ ...)
*T omral_times_wf2  19.4 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:|omral(g;r)|.  ps ** qs ∈ |omral(g;r)|
|
BY (RepD ...a)
|
1. g: OCMon
2. r: CRng
3. ps: |omral(g;r)|
4. qs: |omral(g;r)|
⊢ ps ** qs ∈ |omral(g;r)|
|
BY OnHyps [4;3] AddCarProperties
|  THEN Reduce 0 THEN (MemTypeCD ...)
|\
| 4. ↑sd_ordered(map(λx.x.1;ps))
| 5. ¬↑(0 ∈b map(λx.x.2;ps))
```

```
| 6. qs: |omral(g;r)|
| 7. ↑sd_ordered(map(λx.x.1;qs))
| 8. ¬↑(0 ∈_b map(λx.x.2;qs))
| ⊢ ↑sd_ordered(map(λx.x.1;ps ** qs))
| |
1 BY (BLemma ‘omral_times_sd_ordered‘ ...)
 \
  4. ↑sd_ordered(map(λx.x.1;ps))
  5. ¬↑(0 ∈_b map(λx.x.2;ps))
  6. qs: |omral(g;r)|
  7. ↑sd_ordered(map(λx.x.1;qs))
  8. ¬↑(0 ∈_b map(λx.x.2;qs))
  ⊢ ¬↑(0 ∈_b map(λx.x.2;ps ** qs))
  |
  BY (BLemma ‘omral_times_non_zero_vals‘ ...)
*T lookup_omral_times  98.9 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:|omral(g;r)|. ∀z:|g|.
|    (ps ** qs)[z]
|    = msFor{r↓+gp} x ∈ dom(ps). msFor{r↓+gp} y ∈ dom(qs). when (x * y) =_b z. ps[x] * qs[y]
|
BY (RepD THENM OnVar ‘ps’ MoveToConcl
|   THENM BLemma ‘omralist_ind_a‘ ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| 3. qs: |omral(g;r)|
| 4. z: |g|
| ⊢ ([] ** qs)[z]
| |  = msFor{r↓+gp} x ∈ dom([]). msFor{r↓+gp} y ∈ dom(qs). when (x * y) =_b z. [][x] * qs[y]
| |
1 BY Reduce 0
| |
| ⊢ 0 = 0
| |
1 BY Auto
 \
  1. g: OCMon
  2. r: CRng
  3. qs: |omral(g;r)|
  4. z: |g|
  ⊢ ∀ps:|omral(g;r)|
  |    (ps ** qs)[z]
  |    = msFor{r↓+gp} x ∈ dom(ps). msFor{r↓+gp} y ∈ dom(qs). when (x * y) =_b z. ps[x] * qs[y]
  |    ⇒ (∀x:|g|. ∀y:|r|.
  |          ↑before(x;map(λx.x.1;ps))
  |          ⇒ ¬(y = 0)
  |          ⇒ ((<x, y>::ps) ** qs)[z]
  |             = msFor{r↓+gp} x@0 ∈ dom(<x, y>::ps)
  |                 msFor{r↓+gp} y@0 ∈ dom(qs)
  |                   when (x@0 * y@0) =_b z. (<x, y>::ps)[x@0] * qs[y@0])
  |
  BY (RenameBVars [‘x’,‘kp’;‘x@0’,‘x’;‘y’,‘vp’;‘y@0’,‘y’] 0
  |   THENM RepD ...a)
  |
  5. ps: |omral(g;r)|
  6. (ps ** qs)[z]
     = msFor{r↓+gp} kp ∈ dom(ps)
```

```
       msFor{r↓+gp} vp ∈ dom(qs). when (kp * vp) =b z. ps[kp] * qs[vp]
7. kp: |g|
8. vp: |r|
9. ↑before(kp;map(λkp.kp.1;ps))
10. ¬(vp = 0)
⊢ ((<kp, vp>::ps) ** qs)[z]
|  = msFor{r↓+gp} x ∈ dom(<kp, vp>::ps)
|      msFor{r↓+gp} y ∈ dom(qs). when (x * y) =b z. (<kp, vp>::ps)[x] * qs[y]
|
BY Reduce 0
|
⊢ (<kp,vp>* qs ++ (ps ** qs))[z]
|  = (msFor{r↓+gp} y ∈ dom(qs)
|       when (kp * y) =b z. if kp =b kp then vp else ps[kp] fi  * qs[y])
|    +r (msFor{r↓+gp} x ∈ dom(ps)
|          msFor{r↓+gp} y ∈ dom(qs)
|            when (x * y) =b z. if kp =b x then vp else ps[x] fi  * qs[y])
|
BY % The ifthenelse reduction should be done
|    more intelligently. %
|  (RWN 1 (LemmaC `ite_rw_true`) 0
|   THENM RWH (LemmaC `ite_rw_false`) 0
|   THENM RWH (LemmaC `lookup_omral_plus`) 0 ...a)
|\
| 11. x: |(g↓oset)|
| 12. ↑(x ∈b dom(qs))
| ⊢ ↑(kp =b kp)
| |
1 BY (RW bool_to_propC 0 ...)
|\
| 11. x: |(g↓oset)|
| 12. ↑(x ∈b dom(ps))
| 13. x1: |(g↓oset)|
| 14. ↑(x1 ∈b dom(qs))
| ⊢ ¬↑(kp =b x)
| |
1 BY (FLemma `rng_before_imp_before_all` [9] ...a)
| |
| 15. ↑(∀bx(:|g|) ∈ map(λkp.kp.1;ps). x <b kp)
| |
1 BY % Aaargh! %
| |  Assert ⌈↑(∀bx(:|(g↓oset)|) ∈ map(λkp.kp.1;ps). x <b kp)⌉
| |  THENA (Reduce 0 THEN Auto)
| |  THEN (RWH (LemmaC `ball_char`) (-1)
| |        THENM RW bool_to_propC (-1) ...a)
| |
| 16. ∀x:|(g↓oset)|. ↑(x ∈b map(λkp.kp.1;ps)) ⇒ x < kp
| |
1 BY (RepUnfolds ``mset_mem omral_dom oal_dom mk_mset`` 12
|     THEN RW bool_to_propC 0
|     THENM InstHyp [⌈x⌉] 16
|     THENM RelRST ...)
 \
  ⊢ (<kp,vp>* qs)[z] +r (ps ** qs)[z]
  |  = (msFor{r↓+gp} y ∈ dom(qs). when (kp * y) =b z. vp * qs[y])
  |    +r (msFor{r↓+gp} x ∈ dom(ps)
  |          msFor{r↓+gp} y ∈ dom(qs). when (x * y) =b z. ps[x] * qs[y])
```

```
     |
     BY EqCD
     |\
     | ⊢ +r = +r
     | |
     1 BY Auto
     |\
     | ⊢ (<kp,vp>* qs)[z] = msFor{r↓+gp} y ∈ dom(qs). when (kp * y) =b z. vp * qs[y]
     | |
     1 BY (BLemma `lookup_omral_scale_c` ...)
      \
       ⊢ (ps ** qs)[z]
       |  = msFor{r↓+gp} x ∈ dom(ps). msFor{r↓+gp} y ∈ dom(qs). when (x * y) =b z. ps[x] * qs[y]
       |
       BY Trivial
*T lookup_omral_times_a  0.0 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:|omral(g;r)|. ∀z:|g|.
|    (ps ** qs)[z] = (Σx ∈ dom(ps). Σy ∈ dom(qs). when (x * y) =b z. ps[x] * qs[y])
|
BY Unfold `rng_mssum` 0 THEN Lemma `lookup_omral_times`
*T mset_on_grp_eq  1.8 sec.
⊢ ∀g:OCMon. MSet{g↓set} = MSet{g↓oset}
|
BY (D 0 ...a)
|
1. g: OCMon
⊢ MSet{g↓set} = MSet{g↓oset}
|
BY (Assert ⌈MSet{g↓set} ∈ U⌉ ...a)
|
2. MSet{g↓set} ∈ U
|
BY Repeat (D 1)
   THENM OnCls [0;-1] (Eval ``mset``)
   THENM Trivial
*T mset_inc  1.6 sec.
⊢ ∀g:OCMon. MSet{g↓set} ⊆ MSet{g↓oset}
|
BY (Unfold `subtype` 0
|   THENM RepD ...a)
|
1. g: OCMon
2. x: MSet{g↓set}
⊢ x ∈ MSet{g↓oset}
|
BY OnCls [0;2] (Unfold `mset`)
|
2. x: as,bs:(|(g↓set)| List)//(as ≡(|(g↓set)|) bs)
⊢ x ∈ as,bs:(|(g↓oset)| List)//(as ≡(|(g↓oset)|) bs)
|
BY Repeat (D 1)
|
1. car: U
2. g1: eq:(car → car → B)
       × le:(car → car → B)
       × op:(car → car → car)
       × id:car
```

```
       × (car → car)
3. IsMonoid(|<car, g1>|;*;e) ∧ IsEqFun(|<car, g1>|;=b)
4. Comm(|<car, g1>|;*)
5. Linorder(|<car, g1>|;x,y.↑(x ≤b y))
   ∧ Cancel(|<car, g1>|;|<car, g1>|;*)
   ∧ (∀z:|<car, g1>|. monot(|<car, g1>|;x,y.↑(x ≤b y);λw.z * w))
6. x: as,bs:(|(<car, g1>↓set)| List)//(as ≡(|(<car, g1>↓set)|) bs)
⊢ x ∈ as,bs:(|(<car, g1>↓oset)| List)//(as ≡(|(<car, g1>↓oset)|) bs)
|
BY OnCls [0;-1] Reduce
|
6. x: as,bs:(car List)//(as ≡(car) bs)
⊢ x ∈ as,bs:(car List)//(as ≡(car) bs)
|
BY Trivial
*T mset_inc_a  3.9 sec.
⊢ ∀g:OCMon. MSet{g↓oset} ⊆ MSet{g↓set}
|
BY (Unfold ‘subtype‘ 0
|   THENM RepD ...a)
|
1. g: OCMon
2. x: MSet{g↓oset}
⊢ x ∈ MSet{g↓set}
|
BY OnCls [0;2] (Unfold ‘mset‘)
|
2. x: as,bs:(|(g↓oset)| List)//(as ≡(|(g↓oset)|) bs)
⊢ x ∈ as,bs:(|(g↓set)| List)//(as ≡(|(g↓set)|) bs)
|
BY Repeat (D 1)
|
1. car: 𝕌
2. g1: eq:(car → car → 𝔹)
       × le:(car → car → 𝔹)
       × op:(car → car → car)
       × id:car
       × (car → car)
3. IsMonoid(|<car, g1>|;*;e) ∧ IsEqFun(|<car, g1>|;=b)
4. Comm(|<car, g1>|;*)
5. Linorder(|<car, g1>|;x,y.↑(x ≤b y))
   ∧ Cancel(|<car, g1>|;|<car, g1>|;*)
   ∧ (∀z:|<car, g1>|. monot(|<car, g1>|;x,y.↑(x ≤b y);λw.z * w))
6. x: as,bs:(|(<car, g1>↓oset)| List)//(as ≡(|(<car, g1>↓oset)|) bs)
⊢ x ∈ as,bs:(|(<car, g1>↓set)| List)//(as ≡(|(<car, g1>↓set)|) bs)
|
BY OnCls [0;-1] Reduce
|
6. x: as,bs:(car List)//(as ≡(car) bs)
⊢ x ∈ as,bs:(car List)//(as ≡(car) bs)
|
BY Trivial
*T omral_times_dom  292.1 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps,qs:|omral(g;r)|.  ↑(dom(ps ** qs) ⊆b dom(ps) × dom(qs))
|
BY (RepD ...a)
|
```

```
1. g: OCMon
2. r: CRng
3. ps: |omral(g;r)|
4. qs: |omral(g;r)|
⊢ ↑(dom(ps ** qs) ⊆_b dom(ps) × dom(qs))
|
BY (BLemma ‘mem_bsubmset‘ THENM RepD ...a)
|
5. x: |(g↓set)|
6. ↑(x ∈_b dom(ps ** qs))
⊢ ↑(x ∈_b dom(ps) × dom(qs))
|
BY (OnVar ‘ps’ MoveToConcl
|   THEN BLemma ‘omralist_ind_a‘
|   THENM RepD ...a)
|\
| 3. qs: |omral(g;r)|
| 4. x: |(g↓set)|
| 5. ↑(x ∈_b dom([] ** qs))
| ⊢ ↑(x ∈_b dom([]) × dom(qs))
| |
1 BY Reduce 5
| |
| 5. ↑(x ∈_b 0{g↓oset})
| |
1 BY (RWH (LemmaC ‘mset_mem_char‘) 5 ...a)
|    THEN Eval ‘‘oset_of_ocmon‘‘ 5 THEN Trivial
 \
  3. qs: |omral(g;r)|
  4. x: |(g↓set)|
  5. ps: |omral(g;r)|
  6. ↑(x ∈_b dom(ps ** qs)) ⇒ ↑(x ∈_b dom(ps) × dom(qs))
  7. x1: |g|
  8. y: |r|
  9. ↑before(x1;map(λx.x.1;ps))
  10. ¬(y = 0)
  11. ↑(x ∈_b dom((<x1, y>::ps) ** qs))
  ⊢ ↑(x ∈_b dom(<x1, y>::ps) × dom(qs))
  |
  BY Reduce 11
  |
  11. ↑(x ∈_b dom(<x1,y>* qs ++ (ps ** qs)))
  |
  BY % A good example of monotone reasoning %
  |
  |  (RWH (LemmaC ‘omral_plus_dom‘) 11 ...a)
  |
  11. ↑(x ∈_b dom(<x1,y>* qs) ∪ dom(ps ** qs))
  |
  BY (Fold ‘oset_of_ocmon‘ 11
  |   THENM RWH (LemmaC ‘fset_mem_union‘) 11
  |   THENM RW bool_to_propC 11
  |   THENM D 11 ...a)
  |\
  | 11. ↑(x ∈_b dom(<x1,y>* qs))
  | |
  1 BY (RWH (LemmaC ‘omral_dom_scale‘) 11 ...a)
```

```
| |
| 11. ↑(x ∈_b fs-map(λk'.k' * x1, dom(qs)))
| |
1 BY Unfold `fset_map` 11
| |  THENM (RWH (LemmaC `fset_of_mset_mem`) 11 ...a)
| |
| 11. ↑(x ∈_b msmap{g↓oset,g↓oset}(λk'.k' * x1;dom(qs)))
| |
1 BY (RWH (LemmaC `mset_mem_char`) 11 ...a)
| |
| 11. ↑(∃_b{g↓oset} y ∈ msmap{g↓oset,g↓oset}(λk'.k' * x1;dom(qs)). y =_b x)
| |
1 BY (FLemma `bmsexists_char_a` [11] ...a)
| |
| 12. ↓(∃y:|(g↓oset)|. ↑(y ∈_b msmap{g↓oset,g↓oset}(λk'.k' * x1;dom(qs))) ∧ ↑(y =_b x))
| |
1 BY (D 12 THENM CUnhide
| |   THENM ExRepD ...a)
| |
| 12. y1: |(g↓oset)|
| 13. ↑(y1 ∈_b msmap{g↓oset,g↓oset}(λk'.k' * x1;dom(qs)))
| 14. ↑(y1 =_b x)
| |
1 BY (RW bool_to_propC 14  ...a)
| |
| 14. y1 = x
| |
1 BY (RWH (LemmaC `mset_mem_char`) 0 ...a)
| |
| ⊢ ↑(∃_b{g↓set} y ∈ dom(<x1, y>::ps) × dom(qs). y =_b x)
| |
1 BY (Reduce 0 THEN RenameBVars [`y',`y2'] 0
| |   THENM RW (SweepDnC (RevLemmaC `bmsexists_char_rw`
| |                      ORELSEC LemmaC `assert_of_mon_eq`)) 0
| |   THENM Reduce 0 ...a)
| |
| ⊢ ∃y2:|g|. ↑(y2 ∈_b (mset_inj{g↓oset}(x1) + dom(ps)) × dom(qs)) ∧ y2 = x
| |
1 BY (Reduce 14 THEN Inst [⌜y1⌝] 0 ...)
| |
| 14. y1 = x
| ⊢ ↑(y1 ∈_b (mset_inj{g↓oset}(x1) + dom(ps)) × dom(qs))
| |
1 BY Unfold `mset_prod` 0 THEN Fold `oset_of_ocmon` 0
| |  THEN Reduce 0
| |
| ⊢ ↑(y1
| |  ∈_b (msFor{<MSet{g↓oset},∪,0>} v ∈ dom(qs). mset_inj{g↓oset}(x1 * v))
| |     ∪ (msFor{<MSet{g↓oset},∪,0>} u ∈ dom(ps)
| |          msFor{<MSet{g↓oset},∪,0>} v ∈ dom(qs). mset_inj{g↓oset}(u * v)))
| |
1 BY Unfold `oset_of_ocmon` 0 THEN Fold `mset_prod` 0
| |
| ⊢ ↑(y1
| |  ∈_b (msFor{<MSet{g↓set},∪,0>} v ∈ dom(qs). mset_inj{g↓set}(x1 * v)) ∪ (dom(ps) × dom(qs)))
| |
1 BY (RWH (LemmaC `fset_mem_union`) 0
```

```
| |   THENM RW bool_to_propC 0 THENM Sel 1 (D 0) ...a)
| |
| ⊢ ↑(y1 ∈b msFor{<MSet{g↓set},∪,0>} v ∈ dom(qs). mset_inj{g↓set}(x1 * v))
| |
1 BY Unfold ‘mset_map‘ 13 THEN Reduce 13
| |
| 13. ↑(y1 ∈b msFor{mset_mon{g↓oset}} x ∈ dom(qs). mset_inj{g↓oset}(x * x1))
| |
1 BY (Unfold ‘oset_of_ocmon‘ 13
| |   THEN OnMCls [0;13] (\i.
| |     RWH (LambdaC ⌜λz.y1 ∈b z⌝) i
| |   THENM RWH (LemmaWithC [‘n’,⌜<𝔹,∨b>⌝] ‘dist_hom_over_mset_for‘) i
| |   THENM Reduce i) ...a)
| |\
| | 13. ↑(y1 ∈b msFor{mset_mon{g↓set}} x ∈ dom(qs). mset_inj{g↓set}(x * x1))
| | ⊢ (λz.y1 ∈b z) ∈ MonHom(<MSet{g↓set},∪,0>,<𝔹,∨b>)
| | |
1 2 BY (MemTypeCD ...)
| | |
| | ⊢ IsMonHom{<MSet{g↓set},∪,0>,<𝔹,∨b>}(λz.y1 ∈b z)
| | |
1 2 BY (BLemma ‘mset_union_bor_mon_hom‘ ...)
| |\
| | 13. ↑((λz.y1 ∈b z) (msFor{mset_mon{g↓set}} x ∈ dom(qs). mset_inj{g↓set}(x * x1)))
| | ⊢ (λz.y1 ∈b z) ∈ MonHom(mset_mon{g↓set},<𝔹,∨b>)
| | |
1 2 BY (MemTypeCD ...)
| | |
| | ⊢ IsMonHom{mset_mon{g↓set},<𝔹,∨b>}(λz.y1 ∈b z)
| | |
1 2 BY (BLemma ‘mset_sum_bor_mon_hom‘ ...)
|  \
|   13. ↑(∃b{g↓set} x ∈ dom(qs). y1 ∈b mset_inj{g↓set}(x * x1))
|   ⊢ ↑(∃b{g↓set} v ∈ dom(qs). y1 ∈b mset_inj{g↓set}(x1 * v))
|   |
1   BY (RWH (LemmaC ‘abmonoid_comm‘) 0 ...)
 \
  11. ↑(x ∈b dom(ps ** qs))
  |
  BY % Apply IH %
  |  FHyp 6 [11]
  |
  12. ↑(x ∈b dom(ps) × dom(qs))
  |
  BY % would be more elegant to prove some more monotonicity
  |    lemmas and use them here %
  |
  |  (OnMCls [0;-1] (RWH (LemmaC ‘mset_prod_mem‘)) ...a)
  |
  12. ↑(∃b{g↓set} v ∈ dom(ps). ∃b{g↓set} w ∈ dom(qs). x =b (v * w))
  ⊢ ↑(∃b{g↓set} v ∈ dom(<x1, y>::ps). ∃b{g↓set} w ∈ dom(qs). x =b (v * w))
  |
  BY % Hyp must be transformed first, so squashes can be elimmed %
  |  (RW (SweepDnC (LemmaC ‘bmsexists_char_a_rw‘
  |                  ORELSEC LemmaC ‘assert_of_mon_eq‘)) (-1)
  |   THENM Reduce (-1) ...a)
  |
```

```
    12. ↓(∃v:|g|. ↑(v ∈b dom(ps)) ∧ ↓(∃w:|g|. ↑(w ∈b dom(qs)) ∧ x = v * w))
    |
    BY (D 12 THENM CUnhide THENM ExRepD ...a)
    |
    12. v: |g|
    13. ↑(v ∈b dom(ps))
    14. ↓(∃w:|g|. ↑(w ∈b dom(qs)) ∧ x = v * w)
    |
    BY (D 14 THENM CUnhide
    |   THENM ExRepD ...a)
    |
    14. w: |g|
    15. ↑(w ∈b dom(qs))
    16. x = v * w
    |
    BY % bug in rewrite code causes crash when renaming left out %
    |  RenameBVars [‘v’,‘v1’;‘w’,‘w1’] 0 THEN
    |  (RW (SweepDnC (RevLemmaC ‘bmsexists_char_rw‘
    |                 ORELSEC LemmaC ‘assert_of_mon_eq‘)) 0
    |   THENM Reduce 0 ...a)
    |
    ⊢ ∃v1:|g|
    |   ↑(v1 ∈b mset_inj{g↓oset}(x1) + dom(ps)) ∧ (∃w1:|g|. ↑(w1 ∈b dom(qs)) ∧ x = v1 * w1)
    |
    BY (Inst [⌜v⌝;⌜w⌝] 0 ...)
    |
    ⊢ ↑(v ∈b mset_inj{g↓oset}(x1) + dom(ps))
    |
    BY Fold ‘oset_of_ocmon‘ 0
       THEN Reduce 0 THEN (RW bool_to_propC 0
                           THENM Sel 2 (D 0) ...)
*T omral_times_assoc  648.7 sec.
⊢ ∀g:OCMon. ∀a:CRng.  Assoc(|omral(g;a)|;λps,qs.ps ** qs)
|
BY Force ‘5‘ (Eval ‘‘assoc‘‘ 0)
|  THEN RenameBVars [‘x’,‘ps’;‘y’,‘qs’;‘z’,‘rs’] 0
|  THEN (RepD ...a)
|
1. g: OCMon
2. a: CRng
3. ps: |omral(g;a)|
4. qs: |omral(g;a)|
5. rs: |omral(g;a)|
⊢ ps ** (qs ** rs) = (ps ** qs) ** rs
|
BY (BLemma ‘omral_lookups_same_a‘
|   THENM D 0 ...a)
|
6. u: |g|
⊢ (ps ** (qs ** rs))[u] = ((ps ** qs) ** rs)[u]
|
BY (RWH (LemmaC ‘lookup_omral_times‘) 0 ...a)
|
⊢ msFor{a↓+gp} x ∈ dom(ps)
|    msFor{a↓+gp} y ∈ dom(qs ** rs). when (x * y) =b u. ps[x] * (qs ** rs)[y]
|  = msFor{a↓+gp} x ∈ dom(ps ** qs)
|      msFor{a↓+gp} y ∈ dom(rs). when (x * y) =b u. (ps ** qs)[x] * rs[y]
```

```
|
BY % Expand domains to simplify later ‘when’ cancellation %
|
|  (RWN 2 (LemmaWithC [‘q’,⌈dom(qs) × dom(rs)⌉]
|           ‘mset_for_dom_shift‘) 0
|   THENM
|   RWN 3 (LemmaWithC [‘q’,⌈dom(ps) × dom(qs)⌉]
|           ‘mset_for_dom_shift‘) 0 ...a)
|\
| 7. x: |(g↓oset)|
| 8. ↑(x ∈b dom(ps))
| ⊢ ↑(dom(qs ** rs) ⊆b dom(qs) × dom(rs))
| |
1 BY (BLemma ‘omral_times_dom‘ ...)
|\
| 7. x: |(g↓oset)|
| 8. ↑(x ∈b dom(ps))
| 9. x1: |(g↓oset)|
| 10. ↑(x1 ∈b (dom(qs) × dom(rs)) - dom(qs ** rs))
| ⊢ when (x * x1) =b u. ps[x] * (qs ** rs)[x1] = e
| |
1 BY (Reduce 0 THEN
| |   RWN 2 (LemmaC ‘lookup_omral_eq_zero‘) 0
| |   THENM RW RngNormC 0
| |   THENM RWH (LemmaC ‘rng_when_of_zero‘) 0 ...)
| |
| ⊢ ¬↑(x1 ∈b dom(qs ** rs))
| |
1 BY % Note for when automating rewrite condition solving:
|      This is example of rewrite condition two levels deep %
|    (RWH (LemmaC ‘mset_mem_diff‘) 10
|     THENM RW bool_to_propC 10 ...)
|\
| ⊢ ↑(dom(ps ** qs) ⊆b dom(ps) × dom(qs))
| |
1 BY (BLemma ‘omral_times_dom‘ ...)
|\
| 7. x: |(g↓oset)|
| 8. ↑(x ∈b (dom(ps) × dom(qs)) - dom(ps ** qs))
| ⊢ msFor{a↓+gp} y ∈ dom(rs). when (x * y) =b u. (ps ** qs)[x] * rs[y] = e
| |
1 BY (Reduce 0 THEN
| |   RWN 1 (LemmaC ‘lookup_omral_eq_zero‘) 0
| |   THENM RW RngNormC 0 ...a)
| |\
| | 9. x1: |(g↓oset)|
| | 10. ↑(x1 ∈b dom(rs))
| | ⊢ ¬↑(x ∈b dom(ps ** qs))
| | |
1 2 BY (RWH (LemmaC ‘mset_mem_diff‘) 8
| |     THENM RW bool_to_propC 8 ...)
|  \
|   ⊢ msFor{a↓+gp} y ∈ dom(rs). when (x * y) =b u. 0 = 0
|   |
1   BY (RWH (LemmaC ‘rng_when_of_zero‘) 0
|       THENM RWH (MacroC ‘x‘ IdC ⌈0⌉ ReduceC ⌈e⌉) 0
|       THENM RWH (LemmaC ‘mset_for_of_id‘) 0 ...)
```

```
\
⊢ msFor{a↓+gp} x ∈ dom(ps)
|    msFor{a↓+gp} y ∈ dom(qs) × dom(rs). when (x * y) =b u. ps[x] * (qs ** rs)[y]
|  = msFor{a↓+gp} x ∈ dom(ps) × dom(qs)
|       msFor{a↓+gp} y ∈ dom(rs). when (x * y) =b u. (ps ** qs)[x] * rs[y]
|
BY % Expand lookups of inner multiplications %
|  (RWH (LemmaC ‘lookup_omral_times‘) 0 ...a)
|
⊢ msFor{a↓+gp} x ∈ dom(ps)
|    msFor{a↓+gp} y ∈ dom(qs) × dom(rs)
|       when (x * y) =b u.
|         ps[x]
|         * (msFor{a↓+gp} x ∈ dom(qs)
|              msFor{a↓+gp} y1 ∈ dom(rs). when (x * y1) =b y. qs[x] * rs[y1])
|  = msFor{a↓+gp} x ∈ dom(ps) × dom(qs)
|       msFor{a↓+gp} y ∈ dom(rs)
|         when (x * y) =b u.
|           (msFor{a↓+gp} x1 ∈ dom(ps)
|              msFor{a↓+gp} y ∈ dom(qs). when (x1 * y) =b x. ps[x1] * qs[y])
|           * rs[y]
|
BY Fold ‘rng_mssum‘ 0
|
⊢ (Σx ∈ dom(ps).
|    Σy ∈ dom(qs) × dom(rs).
|     when (x * y) =b u.
|       ps[x] * (Σx ∈ dom(qs). Σy1 ∈ dom(rs). when (x * y1) =b y. qs[x] * rs[y1]))
|  = (Σx ∈ dom(ps) × dom(qs).
|      Σy ∈ dom(rs).
|       when (x * y) =b u.
|         (Σx1 ∈ dom(ps). Σy ∈ dom(qs). when (x1 * y) =b x. ps[x1] * qs[y]) * rs[y])
|
BY % Float up the Sigma’s and when’s %
|  (RWW "rng_times_mssum_l
|        rng_times_mssum_r
|        rng_mssum_when_swap<
|        rng_times_when_l
|        rng_times_when_r" 0 ...a)
|
⊢ (Σx ∈ dom(ps).
|    Σy ∈ dom(qs) × dom(rs).
|     Σx1 ∈ dom(qs).
|      Σy1 ∈ dom(rs). when (x * y) =b u. when (x1 * y1) =b y. ps[x] * (qs[x1] * rs[y1]))
|  = (Σx ∈ dom(ps) × dom(qs).
|      Σy ∈ dom(rs).
|       Σx1 ∈ dom(ps).
|        Σy1 ∈ dom(qs). when (x * y) =b u. when (x1 * y1) =b x. (ps[x1] * qs[y1]) * rs[y])
|
BY % Bring together the Sigma’s and when’s that cancel %
|  (RW (NthsC [2;3]
|       (HereDnC (LemmaC ‘rng_mssum_swap‘
|                  ORELSEC LemmaC ‘rng_when_swap‘))) 0 ...a)
|
⊢ (Σx ∈ dom(ps).
|    Σx1 ∈ dom(qs).
|     Σy1 ∈ dom(rs).
```

```
   |      Σy ∈ dom(qs) × dom(rs).
   |       when (x1 * y1) =b y. when (x * y) =b u. ps[x] * (qs[x1] * rs[y1]))
   |  = (Σy ∈ dom(rs).
   |      Σx1 ∈ dom(ps).
   |       Σy1 ∈ dom(qs).
   |        Σx ∈ dom(ps) × dom(qs).
   |         when (x1 * y1) =b x. when (x * y) =b u. (ps[x1] * qs[y1]) * rs[y])
   |
  BY % Setup and do cancellation %
   |  (
   |   RWNs [1;3] (LemmaC ‘grp_eq_sym‘) 0
   |   THENM RWNs [1;3] oset_of_ocmonC 0
   |   THENM RWH (LemmaC ‘fset_for_when_eq‘) 0 ...a)
   |\
   | 7. x: |(g↓oset)|
   | 8. ↑(x ∈b dom(ps))
   | 9. x1: |(g↓oset)|
   | 10. ↑(x1 ∈b dom(qs))
   | 11. x2: |(g↓oset)|
   | 12. ↑(x2 ∈b dom(rs))
   | ⊢ ↑(x1 * x2 ∈b dom(qs) × dom(rs))
   | |
  1 BY (BLemma ‘prod_in_mset_prod‘ ...)
   |\
   | 7. x: |(g↓oset)|
   | 8. ↑(x ∈b dom(rs))
   | 9. x1: |(g↓oset)|
   | 10. ↑(x1 ∈b dom(ps))
   | 11. x2: |(g↓oset)|
   | 12. ↑(x2 ∈b dom(qs))
   | ⊢ ↑(x1 * x2 ∈b dom(ps) × dom(qs))
   | |
  1 BY (BLemma ‘prod_in_mset_prod‘ ...)
    \
     ⊢ (Σx ∈ dom(ps).
     |     Σx1 ∈ dom(qs). Σy1 ∈ dom(rs). when (x * (x1 * y1)) =b u. ps[x] * (qs[x1] * rs[y1]))
     |  = (Σy ∈ dom(rs).
     |      Σx1 ∈ dom(ps). Σy1 ∈ dom(qs). when ((x1 * y1) * y) =b u. (ps[x1] * qs[y1]) * rs[y])
     |
     BY % Final normalization of l and r sides to same %
        (RWN 3 (HereDnC (LemmaC ‘rng_mssum_swap‘)) 0
         THENM RW MonNormC 0
         THENM RW RngNormC 0 ...)
*T omral_times_assoc_a  0.4 sec.
⊢ ∀g:OCMon. ∀a:CRng. ∀ps,qs,rs:|omral(g;a)|.  ps ** (qs ** rs) = (ps ** qs) ** rs
|
BY AssertLemma ‘omral_times_assoc‘ []
   THENM Force ‘6‘ (Eval ‘‘assoc‘‘ (-1))
   THEN Trivial
#T omral_times_assoc_b  351.6 sec.
⊢ ∀g:OCMon. ∀a:CRng. ∀ps,qs,rs:|omral(g;a)|.  ps ** (qs ** rs) = (ps ** qs) ** rs
|
BY (RepD
|   THENM BLemma ‘omral_lookups_same_a‘
|   THENM RepD ...a)
|
1. g: OCMon
```

```
2. a: CRng
3. ps: |omral(g;a)|
4. qs: |omral(g;a)|
5. rs: |omral(g;a)|
6. u: |g|
⊢ (ps ** (qs ** rs))[u] = ((ps ** qs) ** rs)[u]
|
BY % discard one side to avoid clutter %
|  SplitRel ⌈0⌉
|\
| ⊢ (ps ** (qs ** rs))[u] = 0
| |
1 BY (RWO "lookup_omral_times_a" 0 ...a)
| |
| ⊢ (Σx ∈ dom(ps). Σy ∈ dom(qs ** rs). when (x * y) =b u. ps[x] * (qs ** rs)[y]) = 0
| |
1 BY (RWO "omral_times_dom" 0 ...a)
| |\
| | 7. x: |(g↓oset)|
| | 8. ↑(x ∈b dom(ps))
| | 9. x1: |(g↓oset)|
| | 10. ↑(x1 ∈b (dom(qs) × dom(rs)) - dom(qs ** rs))
| | ⊢ when (x * x1) =b u. ps[x] * (qs ** rs)[x1] = 0
| | |
1 2 BY (RWN 2 (LemmaC ‘lookup_omral_eq_zero‘) 0
| | |   THENM RW RngNormC 0
| | |   THENM RWH (LemmaC ‘rng_when_of_zero‘) 0 ...)
| | |
| | ⊢ ¬↑(x1 ∈b dom(qs ** rs))
| | |
1 2 BY (RWH (LemmaC ‘mset_mem_diff‘) 10
| |     THENM RW bool_to_propC 10 ...)
|  \
|   ⊢ (Σx ∈ dom(ps). Σy ∈ dom(qs) × dom(rs). when (x * y) =b u. ps[x] * (qs ** rs)[y]) = 0
|   |
1   BY (RWW "lookup_omral_times_a" 0 ...a)
|   |
|   ⊢ (Σx ∈ dom(ps).
|   |    Σy ∈ dom(qs) × dom(rs).
|   |     when (x * y) =b u.
|   |       ps[x] * (Σx ∈ dom(qs). Σy1 ∈ dom(rs). when (x * y1) =b y. qs[x] * rs[y1]))
|   |  = 0
|   |
1   BY % float up sigmas and whens %
|   |  (RWW "rng_times_mssum_l
|   |        rng_times_mssum_r
|   |        rng_mssum_when_swap<
|   |        rng_times_when_l
|   |        rng_times_when_r" 0 ...a)
|   |
|   ⊢ (Σx ∈ dom(ps).
|   |    Σy ∈ dom(qs) × dom(rs).
|   |     Σx1 ∈ dom(qs).
|   |      Σy1 ∈ dom(rs). when (x * y) =b u. when (x1 * y1) =b y. ps[x] * (qs[x1] * rs[y1]))
|   |  = 0
|   |
1   BY (RWN 2 (HereDnC (PolyC "rng_mssum_swap rng_when_swap")) 0 ...a)
```

```
|   |
|   ⊢ (Σx ∈ dom(ps).
|   |    Σx1 ∈ dom(qs).
|   |     Σy1 ∈ dom(rs).
|   |      Σy ∈ dom(qs) × dom(rs).
|   |       when (x1 * y1) =b y. when (x * y) =b u. ps[x] * (qs[x1] * rs[y1]))
|   |  = 0
|   |
1   BY (Unfold `oset_of_ocmon` 0 ...a)
|   |
|   ⊢ (Σx ∈ dom(ps).
|   |    Σx1 ∈ dom(qs).
|   |     Σy1 ∈ dom(rs).
|   |      Σy ∈ dom(qs) × dom(rs).
|   |       when (x1 * y1) =b y. when (x * y) =b u. ps[x] * (qs[x1] * rs[y1]))
|   |  = 0
|   |
1   BY (RWN 1 (LemmaC `grp_eq_sym`) 0
|   |   THENM RWH dset_of_monC 0 ...a)
|   |
|   ⊢ (Σx ∈ dom(ps).
|   |    Σx1 ∈ dom(qs).
|   |     Σy1 ∈ dom(rs).
|   |      Σy ∈ dom(qs) × dom(rs).
|   |       when y =b (x1 * y1). when (x * y) =b u. ps[x] * (qs[x1] * rs[y1]))
|   |  = 0
|   |
1   BY (RWO "rng_fset_for_when_eq" 0 ...a)
|   |\
|   | 7. x: |(g↓set)|
|   | 8. ↑(x ∈b dom(ps))
|   | 9. x1: |(g↓set)|
|   | 10. ↑(x1 ∈b dom(qs))
|   | 11. x2: |(g↓set)|
|   | 12. ↑(x2 ∈b dom(rs))
|   | ⊢ ↑(x1 * x2 ∈b dom(qs) × dom(rs))
|   | |
1   2 BY (BLemma `prod_in_mset_prod` ...)
|    \
|     ⊢ (Σx ∈ dom(ps).
|     |    Σx1 ∈ dom(qs). Σy1 ∈ dom(rs). when (x * (x1 * y1)) =b u. ps[x] * (qs[x1] * rs[y1]))
|     |  = 0
|     |
|     INCOMPLETE
 \
  ⊢ 0 = ((ps ** qs) ** rs)[u]
  |
  INCOMPLETE
*T omral_times_comm  52.7 sec.
⊢ ∀g:OCMon. ∀a:CRng.  Comm(|omral(g;a)|;λps,qs.ps ** qs)
|
BY Force `5` (Eval ``comm`` 0)
|  THEN RenameBVars [`x',`ps';`y',`qs'] 0
|  THEN (RepD ...a)
|
1. g: OCMon
2. a: CRng
```

```
3. ps: |omral(g;a)|
4. qs: |omral(g;a)|
⊢ ps ** qs = qs ** ps
|
BY (BLemma ‘omral_lookups_same_a‘
|   THENM D 0 ...a)
|
5. u: |g|
⊢ (ps ** qs)[u] = (qs ** ps)[u]
|
BY (RWH (LemmaC ‘lookup_omral_times‘) 0
|   THENM Fold ‘rng_mssum‘ 0 ...a)
|
⊢ (Σx ∈ dom(ps). Σy ∈ dom(qs). when (x * y) =_b u. ps[x] * qs[y])
|  = (Σx ∈ dom(qs). Σy ∈ dom(ps). when (x * y) =_b u. qs[x] * ps[y])
|
BY % make bvars same in each equand %
|  RenameBVars’ [‘x’,‘y’;‘y’,‘x’] 0
|
|
BY % normalize %
   (RWN 1 (LemmaC ‘rng_mssum_swap‘) 0
    THENM RW CRngNormC 0
    THENM RW AbMonNormC 0 ...)
*T omral_times_comm_a  0.3 sec.
⊢ ∀g:OCMon. ∀a:CRng. ∀ps,qs:|omral(g;a)|.  ps ** qs = qs ** ps
|
BY AssertLemma ‘omral_times_comm‘ []
   THEN Force ‘5‘ (Eval ‘‘comm‘‘ 1)
   THEN Trivial
*T omral_bilinear  155.1 sec.
⊢ ∀g:OCMon. ∀a:CRng.  BiLinear(|omral(g;a)|;λps,qs.ps ++ qs;λps,qs.ps ** qs)
|
BY (RepD THENM BLemma ‘bilinear_comm_elim‘
|   THENM Force ‘5‘ (Reduce 0) ...a)
|\
| 1. g: OCMon
| 2. a: CRng
| ⊢ Comm(|omral(g;a)|;λps,qs.ps ** qs)
| |
1 BY (BLemma ‘omral_times_comm‘ ...)
 \
  1. g: OCMon
  2. a: CRng
  ⊢ ∀a1,x,y:|omral(g;a)|.  a1 ** (x ++ y) = (a1 ** x) ++ (a1 ** y)
  |
  BY (RenameBVars [‘a1’,‘ps’;‘x’,‘qs’;‘y’,‘rs’] 0
  |   THENM RepD
  |   THENM BLemma ‘omral_lookups_same_a‘
  |   THENM D 0 ...a)
  |
  3. ps: |omral(g;a)|
  4. qs: |omral(g;a)|
  5. rs: |omral(g;a)|
  6. u: |g|
  ⊢ (ps ** (qs ++ rs))[u] = ((ps ** qs) ++ (ps ** rs))[u]
  |
```

```
BY % distribute lookup over plus and times %
|  (RWW "lookup_omral_plus lookup_omral_times
|        f:rng_mssum" 0  ...a)
|
⊢ (Σx ∈ dom(ps). Σy ∈ dom(qs ++ rs). when (x * y) =b u. ps[x] * (qs[y] +a rs[y]))
|  = (Σx ∈ dom(ps). Σy ∈ dom(qs). when (x * y) =b u. ps[x] * qs[y])
|    +a (Σx ∈ dom(ps). Σy ∈ dom(rs). when (x * y) =b u. ps[x] * rs[y])
|
BY % equalize domains of summation %
|
|  (RWNs [2;4;6] (LemmaWithC ['q',⌈dom(qs) ∪ dom(rs)⌉]
|           'mset_for_dom_shift') 0
|   THENM Fold 'rng_mssum' 0 ...a)
|\
| 7. x: |(g↓oset)|
| 8. ↑(x ∈b dom(ps))
| ⊢ ↑(dom(qs ++ rs) ⊆b dom(qs) ∪ dom(rs))
| |
1 BY (BLemma 'omral_plus_dom' ...)
|\
| 7. x: |(g↓oset)|
| 8. ↑(x ∈b dom(ps))
| 9. x1: |(g↓oset)|
| 10. ↑(x1 ∈b (dom(qs) ∪ dom(rs)) - dom(qs ++ rs))
| ⊢ when (x * x1) =b u. ps[x] * (qs[x1] +a rs[x1]) = e
| |
1 BY % Have to backtrack a little with lookup distributing %
| |
| |  (Reduce 0
| |   THENM RWH (RevLemmaC 'lookup_omral_plus') 0
| |   THENM RWN 2 (LemmaC 'lookup_omral_eq_zero') 0 ...a)
| |\
| | ⊢ ¬↑(x1 ∈b dom(qs ++ rs))
| | |
1 2 BY (RWH (LemmaC 'mset_mem_diff') 10
| |     THENM RW bool_to_propC 10 ...)
|  \
|   ⊢ when (x * x1) =b u. ps[x] * 0 = 0
|   |
1   BY (RW RngNormC 0
|       THENM RWH (LemmaC 'rng_when_of_zero') 0 ...)
|\
| 7. x: |(g↓oset)|
| 8. ↑(x ∈b dom(ps))
| ⊢ ↑(dom(qs) ⊆b dom(qs) ∪ dom(rs))
| |
1 BY (BLemma 'mem_bsubmset'
|     THENM RepD
|     THENM RWH (LemmaC 'fset_mem_union') 0
|     THENM RW bool_to_propC 0
|     THENM Sel 1 (D 0)  ...)
|\
| 7. x: |(g↓oset)|
| 8. ↑(x ∈b dom(ps))
| 9. x1: |(g↓oset)|
| 10. ↑(x1 ∈b (dom(qs) ∪ dom(rs)) - dom(qs))
| ⊢ when (x * x1) =b u. ps[x] * qs[x1] = e
```

```
  | |
  1 BY (RWH (LemmaC `mset_mem_diff`) 10
  |     THENM RW bool_to_propC 10
  |     THENM Reduce 0
  |     THENM RWN 2 (LemmaC `lookup_omral_eq_zero`) 0
  |     THENM RW RngNormC 0
  |     THENM RWH (LemmaC `rng_when_of_zero`) 0 ...)
  |\
  | 7. x: |(g↓oset)|
  | 8. ↑(x ∈_b dom(ps))
  | ⊢ ↑(dom(rs) ⊆_b dom(qs) ∪ dom(rs))
  | |
  1 BY (BLemma `mem_bsubmset`
  |     THENM RepD
  |     THENM RWH (LemmaC `fset_mem_union`) 0
  |     THENM RW bool_to_propC 0
  |     THENM Sel 2 (D 0)  ...)
  |\
  | 7. x: |(g↓oset)|
  | 8. ↑(x ∈_b dom(ps))
  | 9. x1: |(g↓oset)|
  | 10. ↑(x1 ∈_b (dom(qs) ∪ dom(rs)) - dom(rs))
  | ⊢ when (x * x1) =_b u. ps[x] * rs[x1] = e
  | |
  1 BY (RWH (LemmaC `mset_mem_diff`) 10
  |     THENM RW bool_to_propC 10
  |     THENM Reduce 0
  |     THENM RWN 2 (LemmaC `lookup_omral_eq_zero`) 0
  |     THENM RW RngNormC 0
  |     THENM RWH (LemmaC `rng_when_of_zero`) 0 ...)
   \
   ⊢ (Σx ∈ dom(ps). Σy ∈ dom(qs) ∪ dom(rs). when (x * y) =_b u. ps[x] * (qs[y] +a rs[y]))
   |  = (Σx ∈ dom(ps). Σy ∈ dom(qs) ∪ dom(rs). when (x * y) =_b u. ps[x] * qs[y])
   |    +a (Σx ∈ dom(ps). Σy ∈ dom(qs) ∪ dom(rs). when (x * y) =_b u. ps[x] * rs[y])
   |
   BY % Distribute xa, Sigma and when over + %
      (RWW "rng_times_over_plus.1
            rng_mssum_of_plus
            rng_when_thru_plus" 0  ...)
*T omral_bilinear_a  0.8 sec.
⊢ ∀g:OCMon. ∀a:CRng. ∀ps,qs,rs:|omral(g;a)|.
|    ps ** (qs ++ rs) = (ps ** qs) ++ (ps ** rs) ∧ (qs ++ rs) ** ps = (qs ** ps) ++ (rs ** ps)
|
BY AssertLemma `omral_bilinear` []
   THEN Force `5` (Eval ``bilinear`` 1)
   THEN Trivial
*C omral_one_act_com
                ====================
                OMRAL ONE AND ACTION
                ====================
*D omral_one_df
                Parens ::Prec(preop)::  11<g:g:L>,<r:r:L>== omral_one{}(<g>; <r>)
                11== omral_one{}(<g>; <r>)
*A omral_one                   11 ==  inj(e,1)
*T omral_one_wf  1.1 sec.
⊢ ∀g:OCMon. ∀r:CRng.  11 ∈ |omral(g;r)|
|
```

```
BY (Unfold `omral_one` 0 ...)
*T omral_dom_one  12.4 sec.
⊢ ∀g:OCMon. ∀r:CRng.  ¬(0 = 1) ⇒ dom(11) = mset_inj{g↓oset}(e)
|
BY (RepD THENM RWW "u:omral_one omral_dom_inj" 0 ...a)
|
1. g: OCMon
2. r: CRng
3. ¬(0 = 1)
⊢ if 1 =b 0 then 0{g↓oset} else mset_inj{g↓oset}(e) fi  = mset_inj{g↓oset}(e)
|
BY (SplitOnConclITE ...a)
|\
| 4. 1 = 0
| ⊢ 0{g↓oset} = mset_inj{g↓oset}(e)
| |
1 BY (RelRST ...)
 \
  4. ¬(1 = 0)
  ⊢ mset_inj{g↓oset}(e) = mset_inj{g↓oset}(e)
  |
  BY Auto
*D omral_action_df
               Parens ::Prec(inop)::
                 <v:v:L> ⋅⋅<g:g:L>,<r:r:L> <ps:ps:L>
                 == omral_action{}(<g>; <r>; <v>; <ps>)
               Parens ::Prec(inop)::
                 <v:v:L> ⋅⋅ <ps:ps:L>
                 == omral_action{}(<g>; <r>; <v>; <ps>)
*A omral_action               v ⋅⋅ ps ==  <e,v>* ps
*T omral_action_wf  2.0 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀v:|r|. ∀ps:|omral(g;r)|.  v ⋅⋅ ps ∈ |omral(g;r)|
|
BY (Unfold `omral_action` 0 ...)
*T comb_for_omral_action_wf  1.6 sec.
⊢ (λg,r,v,ps,z.v ⋅⋅ ps) ∈ g:OCMon
|                          → r:CRng
|                          → v:|r|
|                          → ps:|omral(g;r)|
|                          → ↓True
|                          → |omral(g;r)|
|
BY ProveOpCombTyping `omral_action_wf`
*C omral_dom_action_com
               Nice simple example of monotonicity
               reasoning here.
*T omral_dom_action  24.0 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀v:|r|. ∀ps:|omral(g;r)|.  ↑(dom(v ⋅⋅ ps) ⊆b dom(ps))
|
BY (RepD ...a)
|
1. g: OCMon
2. r: CRng
3. v: |r|
4. ps: |omral(g;r)|
⊢ ↑(dom(v ⋅⋅ ps) ⊆b dom(ps))
|
```

```
BY % A nice example of monotonic reasoning %
|
|  (Unfold `omral_action` 0
|   THEN RWH (LemmaC `omral_dom_scale`) 0 ...a)
|
⊢ ↑(fs-map(λk'.k' * e, dom(ps)) ⊆_b dom(ps))
|
BY (BLemma `mem_bsubmset` THENM RepD ...a)
|
5. x: |(g↓oset)|
6. ↑(x ∈_b fs-map(λk'.k' * e, dom(ps)))
⊢ ↑(x ∈_b dom(ps))
|
BY (RWW "u:fset_map fset_of_mset_mem" 6 ...a)
|
6. ↑(x ∈_b msmap{g↓oset,g↓oset}(λk'.k' * e;dom(ps)))
|
BY (RWH (AssertC ⌈(λk'.k' * e) = Id{|(g↓oset)|}⌉) 6
|   THENM RWW "mset_map_id" 6 ...)
|
⊢ (λk'.k' * e) = Id{|(g↓oset)|}
|
BY (Ext THENM Reduce 0 THENM RW MonNormC 0 ...)
*T lookup_omral_action  12.9 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀v:|r|. ∀ps:|omral(g;r)|.  (v ·· ps)[k] = v * ps[k]
|
BY (RepD THENM Unfold `omral_action` 0 ...a)
|
1. g: OCMon
2. r: CRng
3. k: |g|
4. v: |r|
5. ps: |omral(g;r)|
⊢ (<e,v>* ps)[k] = v * ps[k]
|
BY (ReplaceWithEqv
|     MonNormC
|     ⌈(<e,v>* ps)[e * k] = v * ps[k]⌉
|     0 ...a)
|
⊢ (<e,v>* ps)[e * k] = v * ps[k]
|
BY (RWW "lookup_omral_scale_a" 0 ...)
*C omral_alg_com
                =====================================
                ASSEMBLY OF OMRAL FREE MONOID ALGEBRA
                =====================================
*T omral_times_ident_r  2.1 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps:|omral(g;r)|.  ps ** 11 = ps
|
BY (RepD
|   THENM InstLemma `omral_times_comm` [⌈g⌉;⌈r⌉] ...a)
|
1. g: OCMon
2. r: CRng
3. ps: |omral(g;r)|
4. Comm(|omral(g;r)|;λps,qs.ps ** qs)
```

```
⊢ ps ** 11 = ps
|
BY (Force ‘5‘ (Eval ‘‘comm‘‘ (-1))
    THENM RWH (HypC (-1)) 0
    THENM BLemma ‘omral_times_ident_l‘ ...a)
*T omral_times_ident_l  92.8 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps:|omral(g;r)|.  11 ** ps = ps
|
BY (RepD ...a)
|
1. g: OCMon
2. r: CRng
3. ps: |omral(g;r)|
⊢ 11 ** ps = ps
|
BY Unfold ‘omral_one‘ 0
|
⊢ inj(e,1) ** ps = ps
|
BY (BLemma ‘omral_lookups_same_a‘ THENM D 0 ...a)
|
4. u: |g|
⊢ (inj(e,1) ** ps)[u] = ps[u]
|
BY (RWW "lookup_omral_times omral_dom_inj" 0 ...a)
|
⊢ msFor{r↓+gp} x ∈ if 1 =b 0 then 0{g↓oset} else mset_inj{g↓oset}(e) fi
|    msFor{r↓+gp} y ∈ dom(ps). when (x * y) =b u. inj(e,1)[x] * ps[y]
|  = ps[u]
|
BY (SplitOnConclITE ...a)
|\
| 5. 1 = 0
| ⊢ msFor{r↓+gp} x ∈ 0{g↓oset}
| |    msFor{r↓+gp} y ∈ dom(ps). when (x * y) =b u. inj(e,1)[x] * ps[y]
| |  = ps[u]
| |
1 BY Reduce 0
| |
| ⊢ 0 = ps[u]
| |
1 BY (InvertRel 0
|     THENM BLemma ‘ring_triv‘  ...)
 \
  5. ¬(1 = 0)
  ⊢ msFor{r↓+gp} x ∈ mset_inj{g↓oset}(e)
  |    msFor{r↓+gp} y ∈ dom(ps). when (x * y) =b u. inj(e,1)[x] * ps[y]
  |  = ps[u]
  |
  BY (RWW "mset_for_mset_inj" 0
  |   THENM RW MonNormC 0 ...a)
  |
  ⊢ msFor{r↓+gp} y ∈ dom(ps). when y =b u. inj(e,1)[e] * ps[y] = ps[u]
  |
  BY (RWW "lookup_omral_inj" 0 ...)
  |
  ⊢ msFor{r↓+gp} y ∈ dom(ps). when y =b u. (when e =b e. 1) * ps[y] = ps[u]
```

```
  |
  BY (RWN 2 (LemmaC ‘mon_when_true‘) 0
  |   THENM RW RngNormC 0  ...a)
  |\
  | 6. x: |(g↓oset)|
  | 7. ↑(x ∈b dom(ps))
  | ⊢ ↑(e =b e)
  | |
  1 BY (RW bool_to_propC 0 ...)
   \
    ⊢ msFor{r↓+gp} y ∈ dom(ps). when y =b u. ps[y] = ps[u]
    |
    BY (Decide ⌈↑(u ∈b dom(ps))⌉ ...a)
    |\
    | 6. ↑(u ∈b dom(ps))
    | |
    1 BY (RWH oset_of_ocmonC 0
    |     THENM RWW "fset_for_when_eq" 0 ...)
     \
      6. ¬↑(u ∈b dom(ps))
      |
      BY (RWW "mset_for_when_none" 0 ...a)
      |\
      | 7. x: |(g↓oset)|
      | 8. ↑(x ∈b dom(ps))
      | ⊢ ¬↑(x =b u)
      | |
      1 BY (D 0 THENM RW bool_to_propC (-1)
      |     THENM RWW "-1" 8 ...)
       \
        ⊢ e = ps[u]
        |
        BY (Reduce 0
            THENM RWW "lookup_omral_eq_zero" 0 ...)
*T omral_action_one  8.3 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀ps:|omral(g;r)|.  1 ·· ps = ps
|
BY (RepD THENM BLemma ‘omral_lookups_same_a‘ THENM D 0 ...a)
|
1. g: OCMon
2. r: CRng
3. ps: |omral(g;r)|
4. u: |g|
⊢ (1 ·· ps)[u] = ps[u]
|
BY (RWW "lookup_omral_action" 0
    THENM RW RngNormC 0 ...)
*T omral_action_times  11.5 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀v,w:|r|. ∀ps:|omral(g;r)|.  (v * w) ·· ps = v ·· (w ·· ps)
|
BY (RepD THENM BLemma ‘omral_lookups_same_a‘ THENM D 0 ...a)
|
1. g: OCMon
2. r: CRng
3. v: |r|
4. w: |r|
5. ps: |omral(g;r)|
```

```
6. u: |g|
⊢ ((v * w) ·· ps)[u] = (v ·· (w ·· ps))[u]
|
BY (RWW "lookup_omral_action" 0
    THENM RW RngNormC 0 ...)
*T omral_action_times_r1  135.2 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀v:|r|. ∀ps,qs:|omral(g;r)|.  v ·· (ps ** qs) = (v ·· ps) ** qs
|
BY (RepD THENM BLemma ‘omral_lookups_same_a‘ THENM D 0 ...a)
|
1. g: OCMon
2. r: CRng
3. v: |r|
4. ps: |omral(g;r)|
5. qs: |omral(g;r)|
6. u: |g|
⊢ (v ·· (ps ** qs))[u] = ((v ·· ps) ** qs)[u]
|
BY (RWW "lookup_omral_action lookup_omral_times" 0
|   THENM Fold ‘rng_mssum‘ 0 ...a)
|
⊢ v * (Σx ∈ dom(ps). Σy ∈ dom(qs). when (x * y) =_b u. ps[x] * qs[y])
|  = (Σx ∈ dom(v ·· ps). Σy ∈ dom(qs). when (x * y) =_b u. (v * ps[x]) * qs[y])
|
BY (Unfold ‘rng_mssum‘ 0
|   THENM RWH (LemmaC ‘omral_dom_action‘) 0
|   THENM Fold ‘rng_mssum‘ 0 ...a)
|\
| 7. x: |(g↓oset)|
| 8. ↑(x ∈_b dom(ps) - dom(v ·· ps))
| ⊢ msFor{r↓+gp} y ∈ dom(qs). when (x * y) =_b u. (v * ps[x]) * qs[y] = e
| |
1 BY % Fold back action in concl to make proof easy %
| |  (RWN 2 (RevLemmaC ‘lookup_omral_action‘) 0
| |   THENM RWW "mset_mem_diff" 8
| |   THENM RW bool_to_propC 8
| |   THENM RepD ...a)
| |
| 8. ↑(x ∈_b dom(ps))
| 9. ¬↑(x ∈_b dom(v ·· ps))
| ⊢ msFor{r↓+gp} y ∈ dom(qs). when (x * y) =_b u. (v ·· ps)[x] * qs[y] = e
| |
1 BY (RWN 1 (LemmaC ‘lookup_omral_eq_zero‘) 0
| |   THENM Reduce 0 ...a)
| |
| ⊢ msFor{r↓+gp} y ∈ dom(qs). when (x * y) =_b u. 0 * qs[y] = 0
| |
1 BY (RW RngNormC 0
|     THENM RWH (LemmaC ‘rng_when_of_zero‘) 0
|     THENM RWH add_grp_of_rngC 0
|     THENM RWH (LemmaC ‘mset_for_of_id‘) 0
|     THENM Reduce 0  ...)
 \
  ⊢ v * (Σx ∈ dom(ps). Σy ∈ dom(qs). when (x * y) =_b u. ps[x] * qs[y])
  |  = (Σx ∈ dom(ps). Σy ∈ dom(qs). when (x * y) =_b u. (v * ps[x]) * qs[y])
  |
  BY (RWW "rng_times_mssum_l rng_times_when_l" 0
```

```
      THENM RW RngNormC 0 ...)
*T omral_action_times_r2  5.0 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀v:|r|. ∀ps,qs:|omral(g;r)|.  v ·· (ps ** qs) = ps ** (v ·· qs)
|
BY (RepD
    THENM RWH (LemmaC ‘omral_times_comm_a‘) 0
    THENM RWO "omral_action_times_r1<" 0 ...)
*T omral_action_plus_l  13.5 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀v,w:|r|. ∀ps:|omral(g;r)|.  (v +r w) ·· ps = (v ·· ps) ++ (w ·· ps)
|
BY (RepD THENM BLemma ‘omral_lookups_same_a‘ THENM D 0 ...a)
|
1. g: OCMon
2. r: CRng
3. v: |r|
4. w: |r|
5. ps: |omral(g;r)|
6. u: |g|
⊢ ((v +r w) ·· ps)[u] = ((v ·· ps) ++ (w ·· ps))[u]
|
BY (RWW "lookup_omral_action lookup_omral_plus" 0
    THENM RW RngNormC 0 ...)
*T omral_action_plus_r  17.0 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀v:|r|. ∀ps,qs:|omral(g;r)|.  v ·· (ps ++ qs) = (v ·· ps) ++ (v ·· qs)
|
BY (RepD THENM BLemma ‘omral_lookups_same_a‘ THENM D 0 ...a)
|
1. g: OCMon
2. r: CRng
3. v: |r|
4. ps: |omral(g;r)|
5. qs: |omral(g;r)|
6. u: |g|
⊢ (v ·· (ps ++ qs))[u] = ((v ·· ps) ++ (v ·· qs))[u]
|
BY (RWW "lookup_omral_action lookup_omral_plus" 0
    THENM RW RngNormC 0 ...)
*T omral_action_inj  6.9 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k:|g|. ∀v,v’:|r|.  v ·· inj(k,v’) = inj(k,v * v’)
|
BY (RepD THENM BLemma ‘omral_lookups_same_a‘
|   THENM D 0
|   THENM RWW "lookup_omral_action lookup_omral_inj" 0 ...a)
|
1. g: OCMon
2. r: CRng
3. k: |g|
4. v: |r|
5. v’: |r|
6. u: |g|
⊢ v * (when k =b u. v’) = when k =b u. v * v’
|
BY (Fold ‘rng_when‘ 0
    THENM RWW "rng_times_when_l" 0 ...)
*D omral_alg_df               omral_alg(<g:g:*>;<r:r:*>)== omral_alg{}(<g>; <r>)
*A omral_alg   omral_alg(g;r) ==
                 <|omral(g;r)|
```

```
                   , =b
                   , λx,y.tt
                   , λx,y.x ++ y
                   , 00g,r
                   , λx.--x
                   , λx,y.x ** y
                   , 11
                   , λx,y.(inr · )
                   , λa,x.a ·· x>
*T omral_alg_wf  22.5 sec.
⊢ ∀g:OCMon. ∀r:CRng.  omral_alg(g;r) ∈ AlgebraSig(|r|)
|
BY (Unfolds ``omral_alg algebra_sig`` 0 ...)
*T omral_alg_wf2  60.0 sec.
⊢ ∀g:OCMon. ∀r:CRng.  omral_alg(g;r) ∈ r-CAlgebra
|
BY (RepD THENM RepeatM MemTypeCD
|   THENM Force `5` (Reduce 0) ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| ⊢ omral_alg(g;r) ∈ AlgebraSig(|r|)
| |
1 BY Auto
|\
| 1. g: OCMon
| 2. r: CRng
| ⊢ IsGroup(|omral_alg(g;r)|;+omral_alg(g;r);0omral_alg(g;r);-omral_alg(g;r))
| |
1 BY (Assert ⌜oal_grp(g↓oset;r↓+gp) ∈ Group{i}⌝
| |   THENM AddAllProperties (-1)
| |   THENM All (\i.Force `5` (Eval ``oal_grp omral_alg`` i)) ...a)
| |
| 3. <|oal(g↓oset;r↓+gp)|, =b, λx,y.x ≤≤b y, λx,y.x ++ y, 00, λx.--x> ∈ Group{i}
| 4. IsMonoid(|oal(g↓oset;r↓+gp)|;λx,y.x ++ y;00) ∧ IsEqFun(|oal(g↓oset;r↓+gp)|;=b)
| 5. Inverse(|oal(g↓oset;r↓+gp)|;λx,y.x ++ y;00;λx.--x)
| ⊢ IsGroup(|omral(g;r)|;λx,y.x ++ y;00g,r;λx.--x)
| |
1 BY Unfolds ``omralist omral_plus omral_zero omral_minus`` 0
|    THEN (AGenRepD ["compound"] ...)
|\
| 1. g: OCMon
| 2. r: CRng
| ⊢ Comm(|omral_alg(g;r)|;+omral_alg(g;r))
| |
1 BY % Need finer grading of strengths to avoid the
|      ugliness of the first 3 steps here %
|
|    (ARepD ["basic"]
|     THENM RWH AbRedexC 0
|     THENM Force `2` (Reduce 0)
|     THENM BLemma `omral_plus_comm` ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| ⊢ IsAction(|r|;*;1;|omral_alg(g;r)|;·omral_alg(g;r))
| |
```

```
1 BY (AGenRepD ["compound";"basic"] ...a)
| |\
| | 3. a: |r|
| | 4. b: |r|
| | 5. u: |omral_alg(g;r)|
| | ⊢ (a * b) ·omral_alg(g;r) u = a ·omral_alg(g;r) (b ·omral_alg(g;r) u)
| | |
1 2 BY % Can't keep reducing or would go too far... %
| | |  RWH AbRedexC 0 THENM Force `5` (Reduce 0)
| | |
| | ⊢ (a * b) ·· u = a ·· (b ·· u)
| | |
1 2 BY (BLemma `omral_action_times` ...)
|  \
|   3. u: |omral_alg(g;r)|
|   ⊢ 1 ·omral_alg(g;r) u = u
|   |
1   BY RWH AbRedexC 0 THENM Force `5` (Reduce 0)
|   |
|   ⊢ 1 ·· u = u
|   |
1   BY (BLemma `omral_action_one` ...)
|\
| 1. g: OCMon
| 2. r: CRng
| ⊢ IsBilinear(|r|;|omral_alg(g;r)|;|omral_alg(g;r)|;+r;+omral_alg(g;r);+omral_alg(g;r);·omral_
| |
| |  alg(g;r))
| |
1 BY Unfolds ``bilinear_p`` 0
|    THEN RWH AbRedexC 0 THEN Force `5` (Reduce 0)
|    THEN (Backchain ``omral_action_plus_l omral_action_plus_r`` ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| ⊢ IsEqFun(|omral_alg(g;r)|;omral_alg(g;r).eq)
| |
1 BY RWH AbRedexC 0
|    THENM (Assert ⌈omral(g;r) ∈ DSet⌉
|           THENM AddProperties (-1) ...)
|\
| 1. g: OCMon
| 2. r: CRng
| ⊢ IsMonoid(|omral_alg(g;r)|;xomral_alg(g;r);1omral_alg(g;r))
| |
1 BY RWH AbRedexC 0
| |  THENM D 0
| |\
| | ⊢ Assoc(|omral(g;r)|;λx,y.x ** y)
| | |
1 2 BY (BLemma `omral_times_assoc` ...)
|  \
|   ⊢ Ident(|omral(g;r)|;λx,y.x ** y;11)
|   |
1   BY (AGenRepD ["basic"]
|       THENM Force `5` (Reduce 0)
|       THENM Backchain
```

```
|         ‘‘omral_times_ident_l omral_times_ident_r‘‘ ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| ⊢ BiLinear(|omral_alg(g;r)|;+omral_alg(g;r);xomral_alg(g;r))
| |
1 BY RWH AbRedexC 0
|    THENM (BLemma ‘omral_bilinear‘ ...a)
|\
| 1. g: OCMon
| 2. r: CRng
| 3. a: |r|
| ⊢ Dist1op2opLR(|omral_alg(g;r)|;·omral_alg(g;r) a;xomral_alg(g;r))
| |
1 BY RWH AbRedexC 0
|    THENM Force ‘5‘ (Eval ‘‘dist_1op_2op_lr‘‘ 0)
|    THENM (Backchain ‘‘omral_action_times_r1 omral_action_times_r2‘‘ ...a)
 \
  1. g: OCMon
  2. r: CRng
  ⊢ Comm(|omral_alg(g;r)|;xomral_alg(g;r))
  |
  BY (ARepD ["basic"]
      THENM RWH AbRedexC 0
      THENM Force ‘2‘ (Reduce 0)
      THENM BLemma ‘omral_times_comm_a‘ ...a)
*T omral_inj_mon_op  101.8 sec.
⊢ ∀g:OCMon. ∀r:CRng. ∀k,k’:|g|.  inj(k * k’,1) = inj(k,1) ** inj(k’,1)
|
BY (RepD THENM BLemma ‘omral_lookups_same_a‘
|   THENM D 0 ...a)
|
1. g: OCMon
2. r: CRng
3. k: |g|
4. k’: |g|
5. u: |g|
⊢ inj(k * k’,1)[u] = (inj(k,1) ** inj(k’,1))[u]
|
BY (RWW "lookup_omral_times lookup_omral_inj" 0
|   THENM Folds ‘‘rng_when rng_mssum‘‘ 0 ...a)
|
⊢ when (k * k’) =b u. 1
|  = (Σx ∈ dom(inj(k,1)).
|      Σy ∈ dom(inj(k’,1)). when (x * y) =b u. (when k =b x. 1) * (when k’ =b y. 1))
|
BY (RWH (LemmaC ‘omral_dom_inj‘) 0
|   THENM SplitOnConclITE ...a)
|\
| 6. 1 = 0
| ⊢ when (k * k’) =b u. 1
| |  = (Σx@0 ∈ 0{g↓oset}.
| |      Σy ∈ 0{g↓oset}. when (x@0 * y) =b u. (when k =b x@0. 1) * (when k’ =b y. 1))
| |
1 BY Unfold ‘rng_mssum‘ 0
| |  THEN Reduce 0
| |
```

```
| ⊢ when (k * k') =b u. 1 = 0
| |
1 BY (RWW "6 rng_when_of_zero" 0 ...)
 \
  6. ¬(1 = 0)
  ⊢ when (k * k') =b u. 1
  |  = (Σx ∈ mset_inj{g↓oset}(k).
  |       Σy ∈ mset_inj{g↓oset}(k'). when (x * y) =b u. (when k =b x. 1) * (when k' =b y. 1))
  |
  BY (RWW "mset_for_mset_inj" 0 ...a)
  |
  ⊢ when (k * k') =b u. 1 = when (k * k') =b u. (when k =b k. 1) * (when k' =b k'. 1)
  |
  BY (Unfold `rng_when` 0
  |   THEN RWNs [3;4] (LemmaC `mon_when_true`) 0 ...a)
  |\
  | ⊢ ↑(k =b k)
  | |
  1 BY (RW bool_to_propC 0 ...)
  |\
  | ⊢ ↑(k' =b k')
  | |
  1 BY (RW bool_to_propC 0 ...)
   \
    ⊢ when (k * k') =b u. 1 = when (k * k') =b u. 1 * 1
    |
    BY (RW RngNormC 0 ...)
*D omral_alg_umap_df
               alg_umap{<g:mon:*>,<a:rng:*>}(<n:alg:*>,<f:mon->alg:*>)
               == omral_alg_umap{}(<g>; <a>; <n>; <f>)
               alg_umap(<n:alg:*>,<f:mon->alg:*>)== omral_alg_umap{}(<g>; <a>; <n>; <f>)
*A omral_alg_umap             alg_umap(n,f) ==  λps:|omral(g;a)|. Σk ∈ dom(ps). ps[k] ·n (f k)
*T omral_alg_umap_wf  16.0 sec.
⊢ ∀g:OCMon. ∀a:CRng. ∀n:a-Algebra. ∀f:|g| → |n|.  alg_umap(n,f) ∈ |omral(g;a)| → |n|
|
BY (Unfold `omral_alg_umap` 0 ...)
*T omral_alg_umap_is_hom  496.0 sec.
⊢ ∀g:OCMon. ∀a:CRng. ∀n:a-Algebra. ∀f:MonHom(g,n↓rg↓xmn).
|    IsAlgHom{a,omral_alg(g;a),n}(alg_umap(n,f))
|
BY (AGenRepD ["compound";"basic"] ...a)
|\
| 1. g: OCMon
| 2. a: CRng
| 3. n: a-Algebra
| 4. f: MonHom(g,n↓rg↓xmn)
| 5. a1: |omral_alg(g;a)|
| 6. a2: |omral_alg(g;a)|
| ⊢ alg_umap(n,f) (a1 +omral_alg(g;a) a2) = (alg_umap(n,f) a1) +n (alg_umap(n,f) a2)
| |
1 BY All (RW (HigherC AbRedexC))
| |  THENM Force `5` (Eval ``omral_alg_umap`` 0)
| |
| 5. a1: |omral(g;a)|
| 6. a2: |omral(g;a)|
| ⊢ (Σk ∈ dom(a1 ++ a2). (a1 ++ a2)[k] ·n (f k))
| |  = (Σk ∈ dom(a1). a1[k] ·n (f k)) +n (Σk ∈ dom(a2). a2[k] ·n (f k))
```

```
| |
1 BY % First equalize summation domains %
| |  (RWH (LemmaWithC [‘q’,⌈dom(a1) ∪ dom(a2)⌉]
| |          ‘rng_mssum_dom_shift‘) 0 ...a)
| |\
| | ⊢ ↑(dom(a1 ++ a2) ⊆_b dom(a1) ∪ dom(a2))
| | |
1 2 BY (BLemma ‘omral_plus_dom‘ ...)
| |\
| | 7. x: |(g↓oset)|
| | 8. ↑(x ∈_b (dom(a1) ∪ dom(a2)) - dom(a1 ++ a2))
| | ⊢ (a1 ++ a2)[x] ·n (f x) = 0
| | |
1 2 BY (RWW "lookup_omral_eq_zero.2" 0 ...a)
| | |\
| | | ⊢ ¬↑(x ∈_b dom(a1 ++ a2))
| | | |
1 2 3 BY (RWW "fset_mem_union mset_mem_diff" 8
| | |     THENM RW bool_to_propC 8
| | |     THENM ProveProp ...)
| |  \
| |   ⊢ 0 ·n (f x) = 0
| |   |
1 2   BY (RWO "module_act_zero_l" 0
| |       THENM Reduce 0 ...)
| |\
| | ⊢ ↑(dom(a1) ⊆_b dom(a1) ∪ dom(a2))
| | |
1 2 BY (RWW "mem_bsubmset fset_mem_union" 0
| |     THENM RepD
| |     THENM RW bool_to_propC 0
| |     THENM ProveProp ...a)
| |\
| | 7. x: |(g↓oset)|
| | 8. ↑(x ∈_b (dom(a1) ∪ dom(a2)) - dom(a1))
| | ⊢ a1[x] ·n (f x) = 0
| | |
1 2 BY (RWW "lookup_omral_eq_zero module_act_zero_l" 0
| | |   THENM Reduce 0 ...)
| | |
| | ⊢ ¬↑(x ∈_b dom(a1))
| | |
1 2 BY (RWW "fset_mem_union mset_mem_diff" 8
| |     THENM RW bool_to_propC 8
| |     THENM ProveProp ...)
| |\
| | ⊢ ↑(dom(a2) ⊆_b dom(a1) ∪ dom(a2))
| | |
1 2 BY (RWW "mem_bsubmset fset_mem_union" 0
| |     THENM RepD
| |     THENM RW bool_to_propC 0
| |     THENM ProveProp ...a)
| |\
| | 7. x: |(g↓oset)|
| | 8. ↑(x ∈_b (dom(a1) ∪ dom(a2)) - dom(a2))
| | ⊢ a2[x] ·n (f x) = 0
| | |
```

```
1 2 BY (RWW "lookup_omral_eq_zero module_act_zero_l" 0
| | |   THENM Reduce 0 ...)
| | |
| | ⊢ ¬↑(x ∈b dom(a2))
| | |
1 2 BY (RWW "fset_mem_union mset_mem_diff" 8
| |     THENM RW bool_to_propC 8
| |     THENM ProveProp ...)
|  \
|   ⊢ (Σk ∈ dom(a1) ∪ dom(a2). (a1 ++ a2)[k] ·n (f k))
|   |  = (Σk ∈ dom(a1) ∪ dom(a2). a1[k] ·n (f k)) +n (Σk ∈ dom(a1) ∪ dom(a2). a2[k] ·n (f k))
|   |
1   BY (RWW "lookup_omral_plus
|   |        module_act_plus.1" 0 ...a)
|   |
|   ⊢ (Σk ∈ dom(a1) ∪ dom(a2). (a1[k] ·n (f k)) +n (a2[k] ·n (f k)))
|   |  = (Σk ∈ dom(a1) ∪ dom(a2). a1[k] ·n (f k)) +n (Σk ∈ dom(a1) ∪ dom(a2). a2[k] ·n (f k))
|   |
1   BY RWH rng_of_algC 0
|      THENM (RWW "rng_mssum_of_plus<" 0  ...)
|\
| 1. g: OCMon
| 2. a: CRng
| 3. n: a-Algebra
| 4. f: MonHom(g,n↓rg↓xmn)
| 5. u: |a|
| 6. a@0: |omral_alg(g;a)|
| ⊢ alg_umap(n,f) (·omral_alg(g;a) u a@0) = ·n u (alg_umap(n,f) a@0)
| |
1 BY RenameVar ‘a1’ 6
| |  THENM All (RW (HigherC AbRedexC))
| |  THENM Force ‘5‘ (Eval ‘‘omral_alg_umap‘‘ 0)
| |
| 6. a1: |omral(g;a)|
| ⊢ (Σk ∈ dom(u ·· a1). (u ·· a1)[k] ·n (f k)) = ·n u (Σk ∈ dom(a1). a1[k] ·n (f k))
| |
1 BY % Equalize summation domains %
| |  (Unfold ‘rng_mssum‘ 0
| |   THENM RWO "omral_dom_action" 0
| |   THENM Fold ‘rng_mssum‘ 0 ...a)
| |\
| | 7. x: |(g↓oset)|
| | 8. ↑(x ∈b dom(a1) - dom(u ·· a1))
| | ⊢ (u ·· a1)[x] ·n (f x) = e
| | |
1 2 BY Reduce 0
| | |
| | ⊢ (u ·· a1)[x] ·n (f x) = 0n
| | |
1 2 BY (RWW "lookup_omral_eq_zero module_act_zero_l" 0 ...)
| | |
| | ⊢ ¬↑(x ∈b dom(u ·· a1))
| | |
1 2 BY (RWW "fset_mem_union mset_mem_diff" 8
| |     THENM RW bool_to_propC 8
| |     THENM ProveProp ...)
|  \
```

```
|   ⊢ (Σk ∈ dom(a1). (u ·· a1)[k] ·n (f k)) = ·n u (Σk ∈ dom(a1). a1[k] ·n (f k))
|   |
1   BY (RWH (LemmaC `lookup_omral_action`) 0 ...a)
|   |
|   ⊢ (Σk ∈ dom(a1). (u * a1[k]) ·n (f k)) = ·n u (Σk ∈ dom(a1). a1[k] ·n (f k))
|   |
1   BY % n.act is grp hom... Type matching can't figure out n binding:
|   |    because matching requires fill in of forgetful functor. %
|   |
|   |  (RWH (LemmaWithC [`n',⌈n↓rg↓+gp⌉] `dist_hom_over_mset_for`) 0
|   |   THENM Fold `rng_mssum` 0 ...a)
|   |\
|   | ⊢ ·n u ∈ MonHom(n↓rg↓+gp,n↓rg↓+gp)
|   | |
1   2 BY (Fold `grp_of_module` 0
|   |     THENM MemTypeCD
|   |     THEN IfLabL [`set predicate`,
|   |        BLemma `module_act_is_grp_hom`] ...)
|    \
|     ⊢ (Σk ∈ dom(a1). (u * a1[k]) ·n (f k)) = (Σk ∈ dom(a1). ·n u (a1[k] ·n (f k)))
|     |
1     BY (RWW "module_action_p.1" 0 ...a)
|     |
|     ⊢ (Σk ∈ dom(a1). u ·n (a1[k] ·n (f k))) = (Σk ∈ dom(a1). ·n u (a1[k] ·n (f k)))
|     |
1     BY (Unfold `infix_ap` 0 ...)
|\
| 1. g: OCMon
| 2. a: CRng
| 3. n: a-Algebra
| 4. f: MonHom(g,n↓rg↓xmn)
| 5. a1: |omral_alg(g;a)|
| 6. a2: |omral_alg(g;a)|
| ⊢ alg_umap(n,f) (a1 xomral_alg(g;a) a2) = (alg_umap(n,f) a1) xn (alg_umap(n,f) a2)
| |
1 BY All (RW (HigherC AbRedexC))
| |  THEN Force `5` (Eval ``omral_alg_umap`` 0)
| |  THEN RWH rng_to_mod_mssumC 0
| |
| 5. a1: |omral(g;a)|
| 6. a2: |omral(g;a)|
| ⊢ (Σn k ∈ dom(a1 ** a2). (a1 ** a2)[k] ·n (f k))
| |  = (Σn k ∈ dom(a1). a1[k] ·n (f k)) xn (Σn k ∈ dom(a2). a2[k] ·n (f k))
| |
1 BY % Suitably widen summation domain %
| |  (Unfold `mod_mssum` 0
| |   THENM RWO "omral_times_dom" 0
| |   THENM Fold `mod_mssum` 0 ...a)
| |\
| | 7. x: |(g↓oset)|
| | 8. ↑(x ∈_b (dom(a1) × dom(a2)) - dom(a1 ** a2))
| | ⊢ (a1 ** a2)[x] ·n (f x) = e
| | |
1 2 BY (RWW "lookup_omral_eq_zero" 0 ...a)
| | |\
| | | ⊢ ¬↑(x ∈_b dom(a1 ** a2))
| | | |
```

```
1 2 3 BY (RWW "fset_mem_union mset_mem_diff" 8
| | |     THENM RW bool_to_propC 8
| | |     THENM ProveProp ...)
| |  \
| |   ⊢ 0 ·n (f x) = e
| |   |
1 2   BY (Reduce 0
| |       THENM RWW "module_act_zero_l" 0 ...)
|  \
|   ⊢ (Σn k ∈ dom(a1) × dom(a2). (a1 ** a2)[k] ·n (f k))
|   |  = (Σn k ∈ dom(a1). a1[k] ·n (f k)) xn (Σn k ∈ dom(a2). a2[k] ·n (f k))
|   |
1   BY (RWW "lookup_omral_times" 0
|   |   THENM Fold `rng_mssum` 0 ...a)
|   |
|   ⊢ (Σn k ∈ dom(a1) × dom(a2)
|   |     (Σx ∈ dom(a1). Σy ∈ dom(a2). when (x * y) =b k. a1[x] * a2[y]) ·n (f k))
|   |  = (Σn k ∈ dom(a1). a1[k] ·n (f k)) xn (Σn k ∈ dom(a2). a2[k] ·n (f k))
|   |
1   BY % Pull 1st Σn k and when together.
|   |    Would be nice to automate this with some metric-guided rewrites.
|   |    Didn't Bundy write a CADE paper on this a while ago?
|   |  %
|   |  (RWD (PolyC "mod_action_mssum_r mod_action_when_r") 0  ...a)
|   |
|   ⊢ (Σn k ∈ dom(a1) × dom(a2)
|   |     Σn x ∈ dom(a1). Σn y ∈ dom(a2). when (x * y) =b k. (a1[x] * a2[y]) ·n (f k))
|   |  = (Σn k ∈ dom(a1). a1[k] ·n (f k)) xn (Σn k ∈ dom(a2). a2[k] ·n (f k))
|   |
1   BY (RWD (LemmaC `mod_mssum_swap`) 0
|   |   THENM RWH (LemmaC `grp_eq_sym`) 0 ...a)
|   |
|   ⊢ (Σn x ∈ dom(a1)
|   |     Σn y ∈ dom(a2). Σn k ∈ dom(a1) × dom(a2). when k =b (x * y). (a1[x] * a2[y]) ·n (f k))
|   |  = (Σn k ∈ dom(a1). a1[k] ·n (f k)) xn (Σn k ∈ dom(a2). a2[k] ·n (f k))
|   |
1   BY (RWN 3 (UnfoldTopC `mod_mssum`) 0
|   |   THENM RWH dset_of_monC 0
|   |   THENM Unfold `oset_of_ocmon` 0
|   |   THENM RWH (LemmaC `fset_for_when_eq`) 0 ...a)
|   |\
|   | 7. x: |(g↓set)|
|   | 8. ↑(x ∈b dom(a1))
|   | 9. x1: |(g↓set)|
|   | 10. ↑(x1 ∈b dom(a2))
|   | ⊢ ↑(x * x1 ∈b dom(a1) × dom(a2))
|   | |
1   2 BY (BLemma `prod_in_mset_prod` ...)
|    \
|     ⊢ (Σn x ∈ dom(a1). Σn y ∈ dom(a2). (a1[x] * a2[y]) ·n (f (x * y)))
|     |  = (Σn k ∈ dom(a1). a1[k] ·n (f k)) xn (Σn k ∈ dom(a2). a2[k] ·n (f k))
|     |
1     BY (RenameBVars' [`k','x';`k','y'] 0
|     |   THENM RWH (LemmaC `mod_times_mssum_r`) 0
|     |   THENM RWH (LemmaC `mod_times_mssum_l`) 0 ...a)
|     |
|     ⊢ (Σn x ∈ dom(a1). Σn y ∈ dom(a2). (a1[x] * a2[y]) ·n (f (x * y)))
```

```
|     |  = (Σn x ∈ dom(a1). Σn y ∈ dom(a2). (a1[x] ·n (f x)) xn (a2[y] ·n (f y)))
|     |
1     BY (RWH (LemmaC ‘monoid_hom_op‘) 0
|     |   THENM Reduce 0 ...a)
|     |
|     ⊢ (Σn x ∈ dom(a1). Σn y ∈ dom(a2). (a1[x] * a2[y]) ·n (xn (f x) (f y)))
|     |  = (Σn x ∈ dom(a1). Σn y ∈ dom(a2). (a1[x] ·n (f x)) xn (a2[y] ·n (f y)))
|     |
1     BY (RWW "algebra_act_times_lr" 0 ...)
 \
  1. g: OCMon
  2. a: CRng
  3. n: a-Algebra
  4. f: MonHom(g,n↓rg↓xmn)
  ⊢ alg_umap(n,f) 1omral_alg(g;a) = 1n
  |
  BY All (RW (HigherC AbRedexC))
  |  THENM Force ‘5‘ (Eval ‘‘omral_alg_umap‘‘ 0)
  |
  ⊢ (Σk ∈ dom(11). 11[k] ·n (f k)) = 1n
  |
  BY Unfold ‘omral_one‘ 0
  |
  ⊢ (Σk ∈ dom(inj(e,1)). inj(e,1)[k] ·n (f k)) = 1n
  |
  BY (RWH (LemmaC ‘omral_dom_inj‘) 0
  |   THENM SplitOnConclITE ...a)
  |\
  | 5. 1 = 0
  | ⊢ (Σk ∈ 0{g↓oset}. inj(e,1)[k] ·n (f k)) = 1n
  | |
  1 BY Eval ‘‘rng_mssum‘‘ 0
  | |
  | ⊢ 0n = 1n
  | |
  1 BY (InvertRel 0 THENM BLemma ‘module_over_triv_rng‘ ...)
   \
    5. ¬(1 = 0)
    ⊢ (Σk ∈ mset_inj{g↓oset}(e). inj(e,1)[k] ·n (f k)) = 1n
    |
    BY (Unfold ‘rng_mssum‘ 0
    |   THENM RWH (LemmaC ‘mset_for_mset_inj‘) 0 ...a)
    |
    ⊢ inj(e,1)[e] ·n (f e) = 1n
    |
    BY (RWH (LemmaC ‘lookup_omral_inj‘) 0 ...a)
    |
    ⊢ (when e =b e. 1) ·n (f e) = 1n
    |
    BY (RWH (LemmaC ‘mon_when_true‘) 0 ...a)
    |\
    | ⊢ ↑(e =b e)
    | |
    1 BY (RW bool_to_propC 0 ...)
     \
      ⊢ 1 ·n (f e) = 1n
      |
```

```
      BY (RWW "module_action_p.2 monoid_hom_id" 0
          THENM Reduce 0 ...)
*T omral_alg_umap_tri_comm  48.8 sec.
⊢ ∀g:OCMon. ∀a:CRng. ∀n:a-Algebra. ∀f:|g| → |n|.  alg_umap(n,f) o (λk.inj(k,1)) = f
|
BY (RepD THENM New [‘k1’] Ext
|   THENM Eval ‘‘omral_alg_umap‘‘ 0 ...a)
|
1. g: OCMon
2. a: CRng
3. n: a-Algebra
4. f: |g| → |n|
5. k1: |g|
⊢ (Σk ∈ dom(inj(k1,1)). inj(k1,1)[k] ·n (f k)) = f k1
|
BY (RWO "omral_dom_inj" 0 THENM SplitOnConclITE
|   THENM Eval ‘‘rng_mssum‘‘ 0 ...a)
|\
| 6. 1 = 0
| ⊢ 0n = f k1
| |
1 BY (InvertRel 0 THENM BLemma ‘module_over_triv_rng‘ ...)
 \
  6. ¬(1 = 0)
  ⊢ msFor{n↓rg↓+gp} k ∈ mset_inj{g↓oset}(k1). inj(k1,1)[k] ·n (f k) = f k1
  |
  BY (RWW "mset_for_mset_inj" 0 ...a)
  |
  ⊢ inj(k1,1)[k1] ·n (f k1) = f k1
  |
  BY (RWW "lookup_omral_inj mon_when_true" 0 ...a)
  |\
  | ⊢ ↑(k1 =b k1)
  | |
  1 BY (RW bool_to_propC 0 ...)
   \
    ⊢ 1 ·n (f k1) = f k1
    |
    BY (RWW "module_action_p.2" 0 ...)
*T omral_alg_umap_unique  148.2 sec.
⊢ ∀g:OCMon. ∀a:CRng. ∀n:a-Algebra. ∀f:|g| → |n|. ∀f’:a-AlgebraHom(omral_alg(g;a);n).
|    f’ o (λk:|g|. inj(k,1)) = f ⇒ f’ = alg_umap(n,f)
|
BY (RepD THENM New [‘ps’] Ext
|   THENM Eval ‘‘omral_alg_umap‘‘ 0 ...a)
|
1. g: OCMon
2. a: CRng
3. n: a-Algebra
4. f: |g| → |n|
5. f’: a-AlgebraHom(omral_alg(g;a);n)
6. f’ o (λk:|g|. inj(k,1)) = f
7. ps: |omral(g;a)|
⊢ f’ ps = (Σk ∈ dom(ps). ps[k] ·n (f k))
|
BY (RWO "6<" 0 THENM Reduce 0 ...a)
|
```

```
⊢ f' ps = (Σk ∈ dom(ps). ps[k] ·n (f' inj(k,1)))
|
BY (Unfold `rng_mssum` 0
|   THENM Fold `grp_of_module` 0
|   THENM RWO "module_hom_action<" 0 ...a)
|
⊢ f' ps = msFor{n↓grp} k ∈ dom(ps). f' (·omral_alg(g;a) ps[k] inj(k,1))
|
BY (RWH (RevLemmaWithC [`m',⌈omral_alg(g;a)↓grp⌉] `dist_hom_over_mset_for`) 0 ...a)
|\
| ⊢ f' ∈ MonHom(omral_alg(g;a)↓grp,n↓grp)
| |
1 BY (AddAllProperties 5
| |   THENM MemTypeCD
| |   THEN IfLabL [`set predicate`,
| |     BLemma `module_hom_is_grp_hom`]
| |   THENM AGenRepD ["compound";"basic"]
| |   THENM HypBackchain ...a)
| |\
| | 6. ∀a1,a2:|omral_alg(g;a)|.  f' (a1 +omral_alg(g;a) a2) = (f' a1) +n (f' a2)
| | 7. ∀u:|a|. fun_thru_1op(|omral_alg(g;a)|;|n|;·omral_alg(g;a) u;·n u;f')
| | 8. ∀a1,a2:|omral_alg(g;a)|.  f' (a1 xomral_alg(g;a) a2) = (f' a1) xn (f' a2)
| | 9. f' 1omral_alg(g;a) = 1n
| | 10. f' o (λk:|g|. inj(k,1)) = f
| | 11. ps: |omral(g;a)|
| | ⊢ f' ∈ |(omral_alg(g;a)↓grp)| → |(n↓grp)|
| | |
1 2 BY % Inclusion should be fixed to get this %
| |    (RWH AbRedexC 0 ...)
|  \
|   6. ∀a1,a2:|omral_alg(g;a)|.  f' (a1 +omral_alg(g;a) a2) = (f' a1) +n (f' a2)
|   7. ∀u:|a|. fun_thru_1op(|omral_alg(g;a)|;|n|;·omral_alg(g;a) u;·n u;f')
|   8. ∀a1,a2:|omral_alg(g;a)|.  f' (a1 xomral_alg(g;a) a2) = (f' a1) xn (f' a2)
|   9. f' 1omral_alg(g;a) = 1n
|   10. f' o (λk:|g|. inj(k,1)) = f
|   11. ps: |omral(g;a)|
|   12. u: |a|
|   13. a@0: |omral_alg(g;a)|
|   ⊢ f' (·omral_alg(g;a) u a@0) = ·n u (f' a@0)
|   |
1   BY (Unfold `fun_thru_1op` 7
|       THENM HypBackchain ...)
 \
  ⊢ f' ps = f' (msFor{omral_alg(g;a)↓grp} k ∈ dom(ps). ·omral_alg(g;a) ps[k] inj(k,1))
  |
  BY (EqCD ...)
  |
  ⊢ ps = msFor{omral_alg(g;a)↓grp} k ∈ dom(ps). ·omral_alg(g;a) ps[k] inj(k,1)
  |
  BY RWH AbRedexC 0
  |  THENM Force `5` (Reduce 0)
  |
  ⊢ ps = msFor{omral_alg(g;a)↓grp} k ∈ dom(ps). ps[k] ·· inj(k,1)
  |
  BY (RWW "omral_action_inj" 0 ...)
  |
  ⊢ ps = msFor{omral_alg(g;a)↓grp} k ∈ dom(ps). inj(k,ps[k] * 1)
```

```
  |
  BY (RW RngNormC 0
      THENM RWW "omral_fact_a<" 0 ...)
*D omral_fma_df               omral_fma(<g:g:*>;<a:a:*>)== omral_fma{}(<g>; <a>)
*A omral_fma   omral_fma(g;a) ==  <omral_alg(g;a), λk.inj(k,1), λn,f.alg_umap(n,f)>
*T omral_fma_wf  22.4 sec.
⊢ ∀g:OCMon. ∀a:CRng.  omral_fma(g;a) ∈ FMASig(g;a)
|
BY (Unfolds ‘‘omral_fma fma_sig‘‘ 0 ...)
*T omral_fma_wf2  53.0 sec.
⊢ ∀g:OCMon. ∀a:CRng.  omral_fma(g;a) ∈ FMonAlg(g;a)
|
BY (RepD THENM MemTypeCD ...)
|\
| 1. g: OCMon
| 2. a: CRng
| ⊢ IsMonHom{g,omral_fma(g;a).alg↓rg↓xmn}(omral_fma(g;a).inj)
| |
1 BY (AGenRepD ["compound";"basic"]
| |   THENM Force ‘5‘ (Reduce 0) ...a)
| |\
| | 3. a1: |g|
| | 4. a2: |g|
| | ⊢ inj(a1 * a2,1) = inj(a1,1) ** inj(a2,1)
| | |
1 2 BY (BLemma ‘omral_inj_mon_op‘ ...)
|  \
|   ⊢ inj(e,1) = 11
|   |
1   BY (Fold ‘omral_one‘ 0 ...)
 \
  1. g: OCMon
  2. a: CRng
  3. n: a-Algebra
  4. f: MonHom(g,n↓rg↓xmn)
  ⊢ omral_fma(g;a).umap n f = !f’:|omral_fma(g;a).alg| → |n|
  |                              IsAlgHom{a,omral_fma(g;a).alg,n}(f’)
  |                              ∧ f’ o omral_fma(g;a).inj = f
  |
  BY (Unfold ‘uni_sat‘ 0 THEN GenRepD
  |   THENM All (\i.Force ‘5‘ (Reduce i)) ...a)
  |\
  | ⊢ IsAlgHom{a,omral_alg(g;a),n}(alg_umap(n,f))
  | |
  1 BY (BLemma ‘omral_alg_umap_is_hom‘ ...)
  |\
  | ⊢ alg_umap(n,f) o (λk.inj(k,1)) = f
  | |
  1 BY (BLemma ‘omral_alg_umap_tri_comm‘ ...)
   \
    5. a’: |omral(g;a)| → |n|
    6. IsAlgHom{a,omral_alg(g;a),n}(a’)
    7. a’ o (λk.inj(k,1)) = f
    ⊢ a’ = alg_umap(n,f)
    |
    BY (BLemma ‘omral_alg_umap_unique‘ ...)
    |
```

```
    ⊢ a' ∈ a-AlgebraHom(omral_alg(g;a);n)
    |
    BY % Peculiarities of definitions make this a mess. %
    |  (ARepD ["compound";"basic"]
    |   THENM RepeatM MemTypeCD
    |   THEN IfLabL [`set predicate`,AGenRepD ["compound";"basic"]
    |                THENM HypBackchain] ...)
    |
    6. ∀a1,a2:|omral_alg(g;a)|.  a' (a1 +omral_alg(g;a) a2) = (a' a1) +n (a' a2)
    7. ∀u:|a|. fun_thru_1op(|omral_alg(g;a)|;|n|;·omral_alg(g;a) u;·n u;a')
    8. ∀a1,a2:|omral_alg(g;a)|.  a' (a1 xomral_alg(g;a) a2) = (a' a1) xn (a' a2)
    9. a' 1omral_alg(g;a) = 1n
    10. a' o (λk.inj(k,1)) = f
    11. u: |a|
    12. a@0: |omral_alg(g;a)|
    ⊢ a' (·omral_alg(g;a) u a@0) = ·n u (a' a@0)
    |
    BY (Unfold `fun_thru_1op` 7
        THENM HypBackchain ...)
*C polynom_3_end              **********************************
```