TPPmark2014

Fadoua Ghourabi
Kwansei Gakuin University
fadouaghourabi@gmail.com

November 20, 2014

Contents

theory *TPPmark2014*

imports
Main Semiring-Normalization Orderings

begin

lemmas *mult-nat19* = *Semiring-Normalization.comm-semiring-1-class.normalizing-semiring-rules*(*19*)
lemmas *mult-nat-comm* = *comm-semiring-1-class.normalizing-semiring-rules*(*7*)

1 Proof of (i)

lemma *power2-sum*:
fixes *a b*::*nat*
shows (*a* + *b*)ˆ*2* = *a*ˆ*2* + *2*∗*a*∗*b* + *b*ˆ*2*
apply (*subst power2-eq-square*)
apply (*subst add-mult-distrib*)
apply (*subst add-mult-distrib2*)+
apply *auto*
apply (*subst power2-eq-square*[*THEN sym*])+
by *simp*

lemma *power2-mod3*:
fixes *a*::*nat*
shows ¬ (*a*ˆ*2 mod 3* = *2*)

```
proof (rule ccontr, simp)
  assume assm:a^2 mod 3 = 2

  obtain q::nat where q:a^2 div 3 = q by simp
  have apow2:a^2 = 3*q + 2
  apply (subst mod-div-equality2[where ?a = a^2,THEN sym])
  by (subst assm, subst q, simp)

  obtain q0 r0::nat where r0:a mod 3 = r0 and q0:a div 3 = q0 and
 r0less3:r0 < 3 by simp-all
  have a:a = 3*q0 + r0
  apply (subst mod-div-equality2[THEN sym])
  by (subst r0, subst q0, simp)

  with apow2 have (3*q0 + r0)^2 = 3*q + 2 by simp
  then have eq1:(3*q0)^2 + 2*3*q0*r0 + r0^2 = 3*q + 2
  by (subst (asm) power2-sum, simp)

  from r0 have r0 = 0 ∨ r0 = 1 ∨ r0 = 2 by auto
  then show False
  proof
    assume assm1:r0 = 0
    with a have (3*q0)^2 = a^2 by auto
    then have 3*3*q0^2 = a^2
    apply (subst (asm) power2-eq-square, auto)
    apply (subst (asm) power2-eq-square[THEN sym])
    by assumption

    then have b1:3*3*q0^2 = a^2 by auto
    have a^2 mod 3 = 0
    by (subst b1[THEN sym], subst mod-mult-left-eq, auto)

    with assm show False by simp
  next
    assume r0 = 1 ∨ r0 = 2 then show False
    proof
      assume assm2:r0 = 1
      from a have (3*q0)^2 + 2*3*q0 + 1 = a^2
      apply (subst (asm)  assm2)
      apply (erule ssubst)
      by (subst power2-sum, auto)

      then have b2:3*(3*q0^2 + 2*q0) + 1 = a^2
      apply (subst (asm) power2-eq-square)
      by (subst (asm) mult-nat19, auto simp:power2-eq-square)

      have a^2 mod 3 = 1
      apply (subst b2[THEN sym])
      apply (subst Suc-eq-plus1[THEN sym])
```

```
    apply (subst  mod-Suc-eq-Suc-mod)
    by (subst mod-mult-self1-is-0, simp)

    with assm show False by simp
  next
    assume assm3:r0 = 2
    from a have (3*q0)^2 + 2*3*q0*2 + 4 = a^2
    apply (subst (asm)  assm3)
    apply (erule ssubst)
    by (subst power2-sum, auto)

    then have b2:3*(3*q0^2 + 2*q0*2 + 1) + 1 = a^2
    apply (subgoal-tac 4 = 3 +1, auto)
    by (subst (asm) power-mult-distrib, auto)

    have a^2 mod 3 = 1
    apply (subst b2[THEN sym])
    apply (subst Suc-eq-plus1[THEN sym])
    apply (subst mult-nat-comm[of 3  Suc (3 * q0^2 + 2 * q0 * 2)])
    by (subst mod-mult-self3, simp)

    with assm show False by simp
  qed
qed
qed

lemma i:
fixes a::nat
shows (a^2 mod 3 = 0) ∨ (a^2 mod 3 = 1)
by (insert power2-mod3[of a], auto)
```

2 Proof of (ii)

```
lemma three-divides-power2:
fixes a:: nat
assumes 3 dvd (a^2)
shows 3 dvd a
proof -
  from assms have apow2:a*a mod 3 = 0
  apply (subst (asm) dvd-eq-mod-eq-0)
  by (subst power2-eq-square[THEN sym], simp)

  obtain q r where q:a div 3 = q and r:a mod 3 = r by simp-all
  from r have r = 0 ∨ r = 1 ∨ r = 2  by auto
  then show ?thesis
  proof
    assume r = 0
    with r have a mod 3 = 0 by simp
    thus 3 dvd a by auto
```

```
  next
    assume r = 1 ∨ r = 2
    then show 3 dvd a
    proof
      assume r = 1
      with q r have a:a = 3*q + 1 by (metis mod-div-equality2)

      then have a:a^2 = 3*(3*q^2 + 2*q) + 1
      by (rule ssubst, subst power2-sum, auto simp:power2-eq-square)

      have a^2 mod 3 = 1
      apply (subst a)
      apply (subst Suc-eq-plus1[THEN sym])
      apply (subst  mod-Suc-eq-Suc-mod)
      by (subst mod-mult-self1-is-0, simp)

      with apow2 have False by (subst (asm) power2-eq-square, simp)
      thus 3 dvd a by simp
   next
      assume r = 2
      with q r have a:a = 3*q + 2 by (metis mod-div-equality2)

      then have a:a^2 = 3*(3*q^2 + 2*q*2+ 1) + 1
      apply (rule ssubst)
      by (subst power2-sum, auto simp:power2-eq-square)

      have a^2 mod 3 = 1
      apply (subst a)
      apply (subst Suc-eq-plus1[THEN sym])
      apply (subst comm-semiring-1-class.normalizing-semiring-rules(7)[of 3  Suc
(3 * q^2 + 2 * q * 2)])
      by (subst mod-mult-self3, simp)

      with apow2 have False by (subst (asm) power2-eq-square, simp)
      thus 3 dvd a by simp
    qed
  qed
qed

lemma ii:
fixes a b c::nat
assumes a^2 + b^2 = 3*c^2
shows 3 dvd a ∧ 3 dvd b ∧ 3 dvd c
proof (auto)
  from assms have 3 dvd (a^2 + b^2) by auto
  then have ab:((a^2 mod 3) + (b^2 mod 3)) mod 3 = 0
  by (subst (asm) dvd-eq-mod-eq-0, subst (asm) mod-add-eq, simp)

  have amod3:a^2 mod 3 = 0 ∨ a^2 mod 3 = 1 by (rule i)
```

```
  moreover have bmod3:b^2 mod 3 = 0 ∨ b^2 mod 3 = 1 by (rule i)
  ultimately have a^2 mod 3 + b^2 mod 3 = 0 ∨ a^2 mod 3 + b^2 mod 3 = 1
∨ a^2 mod 3 + b^2 mod 3 = 2
  by auto
  then have a^2 mod 3 + b^2 mod 3 = 0
  proof
    assume a^2 mod 3 + b^2 mod 3 = 0
    thus ?thesis by simp
  next
    assume a² mod 3 + b² mod 3 = 1 ∨ a² mod 3 + b² mod 3 = 2
    then have ((a^2 mod 3) + (b^2 mod 3)) mod 3 = 1 ∨ ((a^2 mod 3) + (b^2
mod 3)) mod 3 = 2
    by auto
    with ab have False by simp
    thus ?thesis by simp
  qed

  with amod3 bmod3 have a^2 mod 3 = 0 ∧ b^2 mod 3 = 0
  by auto
 then have a2:3 dvd (a^2) and b2:3 dvd (b^2) by (simp-all add:dvd-eq-mod-eq-0)

  from a2 show threedva:3 dvd a by (rule three-divides-power2)
  from b2 show threedvb:3 dvd b by (rule three-divides-power2)

  from threedva obtain q1 where q1:a = 3*q1 by (metis dvdE)
  from threedvb obtain q2 where q2:b = 3*q2 by (metis dvdE)
  with q1 assms have (3*q1)^2 + (3*q2)^2 = 3*c^2 by auto
  then have 3*(3*q1^2 + 3*q2^2) = 3*c^2
  apply (subst (asm) power-mult-distrib)
  apply (subst (asm) power-mult-distrib)
  by auto

  then have 3*q1^2 + 3*q2^2 = c^2 by (simp)
  then have 3 dvd c^2
  apply (subst (asm) add-mult-distrib2[THEN sym])
  by (erule subst, simp)

  thus threedvc:3 dvd c by (rule three-divides-power2)
qed
```

3 Proof of (iii)

```
lemma div3:
fixes a b c::nat
assumes a^2 + b^2 = 3*c^2
shows (a div 3)^2 + (b div 3)^2 = 3*(c div 3)^2
proof -

  from assms have a:3 dvd a and b:3 dvd b and c:3 dvd c using ii
```

by (*simp-all*)

from *a b c assms* **have** $(3*(a\ div\ 3))\hat{\ }2 + (3*(b\ div\ 3))\hat{\ }2 = 3*(3*(c\ div\ 3))\hat{\ }2$

by (*metis dvd-mult-div-cancel*)

then have $9*(a\ div\ 3)\hat{\ }2 + 9*(b\ div\ 3)\hat{\ }2 = 9*(3*(c\ div\ 3)\hat{\ }2)$
by (*auto simp*:*power-mult-distrib*)

thus *?thesis* **by** *auto*

qed

lemma *iii-c-is-null*:
fixes *a b c*::*nat*
assumes $a\hat{\ }2+b\hat{\ }2=3*c\hat{\ }2$
shows $c = 0$
proof (*rule ccontr*)
assume $c{:}c \neq 0$
let $?Sc = \{z{::}nat.\ 0 < z \wedge (\exists\, x\ y{::}nat.\ x\hat{\ }2+y\hat{\ }2=3*z\hat{\ }2)\}$
from *assms c* **have** $c{:}c \in\ ?Sc$ **by** *auto*
then obtain *cmin*::*nat* **where** $cmin{:}cmin = (LEAST\ x.\ x \in\ ?Sc)$ **by** *auto*

{ fix *z* **assume** $z \in\ ?Sc$
with *cmin* **have** $cminlessz{:}cmin \leq z$
using $Least\text{-}le[of\ \lambda u.\ 0 < u \wedge (\exists\, x\ y.\ x^2 + y^2 = 3 * u^2)\ z]$
by *auto*
} note *res = this*
with *c cmin* **have** $cmininSc{:}cmin \in\ ?Sc$
using $LeastI[of\ \lambda u.\ 0 < u \wedge (\exists\, x\ y.\ x^2 + y^2 = 3 * u^2)\ c]$ **by** *auto*

then have $cminpos{:}0 < cmin$ **by** *auto*
from *cmininSc* **obtain** *a0 b0* **where** $sum{:}a0\hat{\ }2 + b0\hat{\ }2 = 3*cmin\hat{\ }2$ **by** *auto*
then have *3 dvd cmin* **using** *ii* **by** *auto*
then have $cmin = 3*(cmin\ div\ 3)$
apply (*subst mult.commute*)
by (*erule dvd-div-mult-self*[*THEN sym*])
with *cminpos* **have** $cmindiv3{:}0 < (cmin\ div\ 3)$ **by** *simp*
from *sum* **have** $(a0\ div\ 3)\hat{\ }2 + (b0\ div\ 3)\hat{\ }2 = 3*(cmin\ div\ 3)\hat{\ }2$ **using** *div3* **by** *auto*
with *cmindiv3* **have** $a1{:}(cmin\ div\ 3) \in\ ?Sc$ **by** *auto*
from *cminpos* **have** $a2{:}cmin\ div\ 3 < cmin$ **using** *int-div-less-self* **by** *auto*
from *res a1* **have** $cmin \leq (cmin\ div\ 3)$ **by** *simp*

with *a2* **show** *False* **by** *simp*
qed

lemma *iii-ab-are-null*:
fixes *a b c*::*nat*

assumes $a\hat{\ }2 + b\hat{\ }2 = 3*c\hat{\ }2$
shows $a = 0 \wedge b = 0$
proof $-$
  from *assms* **have** $c = 0$ **using** *iii-c-is-null* **by** *simp*
  with *assms* **have** $a\hat{\ }2 + b\hat{\ }2 = 0$ **by** *simp*
  thus $a = 0 \wedge b = 0$ **by** *auto*
qed

end