Soundness of the Quasi-Synchronous Abstraction

Guillaume Baudart and Timothy Bourke

Contents

1 Administrative Lemmas

We prove here general lemmas that will be needed in the following.

lemma *ntimes-suc-dist* [*simp*]:
  fixes T :: *real*
  shows (*Suc j*) $*$ T = $j * T + T$
  by (*simp add*: *comm-semiring-class.distrib real-of-nat-Suc*)

lemma *ntimes-nzero-min* [*intro*]:
  fixes T :: *real* **and** k::*nat*
  shows $[\![\, 0 \leq T; \, 0 < k \,]\!] \Longrightarrow T \leq k * T$
  by (*metis Suc-leI monoid-mult-class.mult.right-neutral*
  mult.commute mult-left-mono real-of-nat-le-iff real-of-nat-one)

lemma *ntimes-bound*:
  fixes C :: *real*
  assumes $C > 0$
  shows $\exists\ k$ :: *nat*. $k * C > D$
  using *assms* **by** (*simp add*: *reals-Archimedean3*)

lemma *card1-x*:
  assumes *card* $S = 1$
  and $x \in S$
  shows $S = \{x\}$
  proof −
    from ‹*card* $S = 1$› ‹$x \in S$› **have** *card* $(S - \{x\}) = 0$
      by (*metis One-nat-def card-Diff-singleton card-infinite diff-Suc-1 zero-neq-one*)
    moreover from ‹*card* $S = 1$› **have** *finite S*
      by (*metis One-nat-def Suc-not-Zero card-infinite*)
    ultimately have $S - \{x\} = \{\}$ **by** *simp*
    thus *?thesis*
      using ‹$x \in S$› **by** (*metis insert-Diff-single insert-absorb*)
  qed

lemma *card2-xyz*:
  assumes *card* $S = 2$
  and $x \in S$
  and $y \in S$
  and $z \in S$
  and $x \neq z$
  and $y \neq z$
  shows $x = y$
  proof −
    from ‹*card* $S = 2$› ‹$x \in S$› **have** *card* $(S - \{x\}) = 1$
      by (*metis Suc-1 Suc-not-Zero card-Diff-singleton card-infinite diff-Suc-1*)
    moreover from ‹$z \in S$› ‹$x \neq z$› **have** $z \in S - \{x\}$
      by *simp*
    ultimately have $S - \{x\} = \{z\}$
      by (*simp add*: *card1-x*)
    moreover from ‹$x \in S$› **have** $S = \{x\} \cup (S - \{x\})$
      by *auto*
    ultimately have $S = \{x, z\}$

by *auto*
thus *?thesis*
using ‹$y \in S$› ‹$y \neq z$› **by** *simp*
qed

lemma *finite-max*:
fixes f :: - $\Rightarrow$ *nat*
assumes *finite* A
and $A \neq \{\}$
shows $\exists\ x \in A.\ f\ x = Max\ \{f\ y \mid y.\ y \in A\}$
using *assms*(*1*−*2*) **proof** (*induct*)
case *insert*
fix x F
assume *finite* F
and $x \notin F$
and $F \neq \{\} \Longrightarrow \exists x{\in}F.\ f\ x = Max\ \{f\ y \mid y.\ y \in F\}$
and *insert* x $F \neq \{\}$
thus $\exists\ z \in insert\ x\ F.\ f\ z = Max\ \{f\ y \mid y.\ y \in insert\ x\ F\}$
proof (*cases* $F = \{\}$)
assume $F = \{\}$
hence $Max\ \{f\ y \mid y.\ y \in insert\ x\ F\} = Max\ \{f\ y \mid y.\ y \in \{x\}\}$
by *simp*
hence $Max\ \{f\ y \mid y.\ y \in insert\ x\ F\} = f\ x$
by *simp*
moreover have $x \in insert\ x\ F$
by *simp*
ultimately have $f\ x = Max\ \{f\ y \mid y.\ y \in insert\ x\ F\}$
by *simp*
thus *?thesis* **by** *simp*
next
assume $\neg\ F = \{\}$
with ‹($F \neq \{\} \Longrightarrow \exists x{\in}F.\ f\ x = Max\ \{f\ y \mid y.\ y \in F\}$)›
obtain z
where $z \in F$
and $f\ z = Max\ \{f\ y \mid y.\ y \in F\}$
by *auto*
thus *?thesis*
proof (*cases* $f\ x > f\ z$)
assume $f\ x > f\ z$
with ‹$f\ z = Max\ \{f\ y \mid y.\ y \in F\}$›
have $f\ x = max\ (f\ x)\ (Max\ \{f\ y \mid y.\ y \in F\})$
by *simp*
also have ... $= Max\ (insert\ (f\ x)\ \{f\ y \mid y.\ y \in F\})$
proof (*rule Max-insert* [*symmetric*])
from ‹*finite* F› **show** *finite* $\{f\ y \mid y.\ y \in F\}$
by *simp*
next
from ‹$\neg\ F = \{\}$› **show** $\{f\ y \mid y.\ y \in F\} \neq \{\}$
by *simp*
qed
also have ... $= Max\ (f\ `\ insert\ x\ F)$
by *simp* (*metis Collect-mem-eq image-Collect*)
finally have $f\ x = Max\ (f\ `\ insert\ x\ F)$ **.**

thus $\exists\, z \in insert\ x\ F.\ f\ z = Max\ \{f\ y\ |y.\ y \in insert\ x\ F\}$
by (*metis Collect-mem-eq image-Collect insertI1*)
next
assume $\neg\ f\ x > f\ z$
with $\langle f\ z = Max\ \{f\ y\ |y.\ y \in F\}\rangle$
have $f\ z = max\ (f\ x)\ (Max\ \{f\ y\ |y.\ y \in F\})$
by *simp*
also have ... $= Max\ (insert\ (f\ x)\ \{f\ y\ |y.\ y \in F\})$
proof (*rule Max-insert* [*symmetric*])
from $\langle finite\ F\rangle$ **show** $finite\ \{f\ y\ |y.\ y \in F\}$
by *simp*
next
from $\langle\neg\ F = \{\}\rangle$ **show** $\{f\ y\ |y.\ y \in F\} \neq \{\}$
by *simp*
qed
also have ... $= Max\ (f\ \text{`}\ insert\ x\ F)$
by *simp* (*metis Collect-mem-eq image-Collect*)
finally have $f\ z = Max\ (f\ \text{`}\ insert\ x\ F)$ **.**
thus $\exists\, z \in insert\ x\ F.\ f\ z = Max\ \{f\ y\ |y.\ y \in insert\ x\ F\}$
using $\langle z \in F\rangle$ **by** (*metis Collect-mem-eq image-Collect insert-iff*)
qed
qed
qed (*simp*)

2 Preamble

In this section, we give the definitions of quasi-periodic systems and formalize the happened before relation. Then we prove general lemmas on this relation.

2.1 Global definitions

typedecl *node*

type-synonym *time* = *real*
type-synonym *delay* = *real*
type-synonym *communication* = *node* ⇒ *node* ⇒ *bool*

record *event* =
node :: *node*
act :: *nat*

syntax
-event :: [*node*, *nat*] ⇒ *event* (-•- [*1000*, *1000*] *1000*)
translations
-event n i ⇌ (| *node* = *n*, *act* = *i* |)

record *tevent* =
date :: *time*
trans :: *delay*

type-synonym *trace* = *event* ⇒ *tevent*

fun *arrival* :: *tevent* ⇒ *time*
  where *arrival te* = *date te* + *trans te*

fun *step* :: *event* ⇒ *event*
  where *step e* = *e*⦇ *act* := *act e* + *1* ⦈

inductive *hb1* :: *trace* ⇒ *event* ⇒ *event* ⇒ *bool*
  for *t* :: *trace*
  where
    hb-subsequent: ⋀*A B i j*. ⟦ *A* = *B*; *i* < *j* ⟧ ⟹ *hb1 t A*•*i B*•*j*
  | *hb-arrival*: ⋀*e1 e2*. ⟦ *arrival* (*t e1*) ≤ *date* (*t e2*) ⟧ ⟹ *hb1 t e1 e2*

lemmas *hb1.cases* [*cases del*]
lemma *hb1-cases* [*elim*!]:
  hb1 t A•*i B*•*j* ⟹
   (*A* = *B* ⟹ *i* < *j* ⟹ *P*) ⟹
   (*arrival* (*t A*•*i*) ≤ *date* (*t B*•*j*) ⟹ *P*) ⟹ *P*
  by (*erule hb1.cases*) *auto*

syntax
  -hb1 :: [*event*, *trace*, *event*] ⇒ *bool* (- ↦- - [*100*, *100*, *100*] *100*)
translations
  -hb1 e1 t e2 ⇌ (*CONST hb1 t e1 e2*)

syntax
  -hb :: [*event*, *trace*, *event*] ⇒ *bool* (- →- - [*100*, *100*, *100*] *100*)
translations
  -hb e1 t e2 ⇌ (*CONST hb1 t*) ^++ *e1 e2*

definition *concur* :: *trace* ⇒ *event* ⇒ *event* ⇒ *bool*
  where *concur t e1 e2* ≡ ¬ (*e1* →*t e2*) ∧ ¬ (*e2* →*t e1*)

syntax
  -concur :: [*event*, *trace*, *event*] ⇒ *bool* (- ∥- - [*100*, *100*, *100*] *100*)
translations
  -concur e1 t e2 ⇌ (*CONST concur t e1 e2*)

2.2 Quasi-periodic system

locale *quasiperiodic-system* =
  fixes *nodes* :: *node set*
  and T_{min} :: *time*
  and T_{max} :: *time*
  and τ_{min} :: *time*
  and τ_{max} :: *time*
  and *com* :: *communication*

  assumes *finnode*: *finite nodes*
  and *node-coherent*: ∀ e:: *event*. *node e* ∈ *nodes*
  and *Tminpos*: $0 < T_{min}$
  and *Tbounds*: $T_{min} \leq T_{max}$
  and *tauminpos*: $0 < \tau_{min}$

and *taubounds*: $\tau_{min} \leq \tau_{max}$
and *com-refl*: $\forall\ N.\ com\ N\ N$
begin

lemma *Tminpos'*: $0 \leq T_{min}$
using *Tminpos* **by** *simp*

lemma *Tmaxpos* : $0 < T_{max}$
using *Tbounds Tminpos* **by** *simp*

lemma *Tmaxpos'* : $0 \leq T_{max}$
using *Tbounds Tminpos'* **by** *simp*

lemma *tauminpos'*: $0 \leq \tau_{min}$
using *tauminpos* **by** *simp*

lemma *taumaxpos* : $0 < \tau_{max}$
using *taubounds tauminpos* **by** *simp*

lemma *taumaxpos'* : $0 \leq \tau_{max}$
using *taubounds tauminpos'* **by** *simp*

definition *quasiperiodic* :: *trace* $\Rightarrow$ *bool*
where *quasiperiodic* $t = (\forall e.$
$0 \leq date\ (t\ e)$
$\wedge\ T_{min} \leq date\ (t\ (step\ e)) - date\ (t\ e)$
$\wedge\ date\ (t\ (step\ e)) - date\ (t\ e) \leq T_{max}$
$\wedge\ \tau_{min} \leq trans\ (t\ e) \wedge trans\ (t\ e) \leq \tau_{max})$

lemma *qp-step*:
assumes *quasiperiodic t*
shows $T_{min} \leq date\ (t\ (step\ e)) - date\ (t\ e)$
$\wedge\ date\ (t\ (step\ e)) - date\ (t\ e) \leq T_{max}$
using *assms* **unfolding** *quasiperiodic-def* **by** (*rule allE* [**where** $x{=}e$]) *simp*

lemma *qp-suc*:
assumes *quasiperiodic t*
shows $T_{min} \leq date\ (t\ A{\bullet}(Suc\ i)) - date\ (t\ A{\bullet}i)$
$\wedge\ date\ (t\ A{\bullet}(Suc\ i)) - date\ (t\ A{\bullet}i) \leq T_{max}$
using *qp-step* [*OF assms*, **where** $e{=}A{\bullet}i$] **by** *simp*

lemmas *qp-suc-min* = *qp-suc* [*THEN conjunct1*]
and *qp-suc-max* = *qp-suc* [*THEN conjunct2*]

lemma *qp-trans*:
assumes *quasiperiodic t*
shows $\tau_{min} \leq trans\ (t\ e) \wedge trans\ (t\ e) \leq \tau_{max}$
using *assms* **unfolding** *quasiperiodic-def* **by** (*rule allE* [**where** $x{=}e$]) *simp*

lemma *qp-cone-lower*:
assumes *quasiperiodic t*
shows $k * T_{min} \leq date\ (t\ A{\bullet}(i + k)) - date\ (t\ A{\bullet}i)$

proof (*induct k*)
  fix *k*
  assume $k * T_{min} \leq$ *date* (*t A*•(*i* + *k*)) − *date* (*t A*•*i*) (**is** - ≤ *?fnik* − *?fni*)
  with ‹*quasiperiodic t*›
    have $T_{min} + k * T_{min} \leq$ *date* (*t A*•(*Suc* (*i* + *k*))) − *?fnik* + (*?fnik* − *?fni*)
      by (*rule add-mono* [*OF qp-suc-min*])
  thus (*Suc k*) $* T_{min} \leq$ *date* (*t A*•(*i* + *Suc k*)) − *date* (*t A*•*i*)
     by (*subst ntimes-suc-dist*) *simp*
qed (*simp*)

lemma *qp-cone-lower-tmin*:
  assumes *qp*: *quasiperiodic t*
  and $i < j$
  shows *date* (*t A*•*i*) + $T_{min} \leq$ *date* (*t A*•*j*)
  proof −
    from ‹$i < j$› **obtain** *k*
    where $j = i + k$
    and $0 < k$
      by (*metis less-imp-add-positive*)
    from *this*(*2*) **have** $T_{min} \leq\ k * T_{min}$
      by (*rule ntimes-nzero-min* [*OF Tminpos'*])
    hence *date* (*t A*•*i*) + $T_{min} \leq$ *date* (*t A*•*i*) + $k * T_{min}$
      by (*rule add-left-mono*)
    also have *date* (*t A*•*i*) + $k * T_{min} \leq$ *date* (*t A*•(*i*+*k*))
      using *qp-cone-lower* [*OF qp*, **where** *k*=*k* **and** *i*=*i* **and** *A*=*A*] **by** *auto*
    finally show *date* (*t A*•*i*) + $T_{min} \leq$ *date* (*t A*•*j*)
      using ‹$j = i + k$› **by** *simp*
  qed

lemma *qp-cone-upper*:
  assumes *quasiperiodic t*
  shows *date* (*t A*•(*i* + *k*)) − *date* (*t A*•*i*) $\leq k * T_{max}$
  proof (*induct k*)
    fix *k*
    assume *date* (*t A*•(*i* + *k*)) − *date* (*t A*•*i*) $\leq\ k * T_{max}$ (**is** *?fnik* − *?fni* ≤ -)
    with ‹*quasiperiodic t*›
      have *date* (*t A*•(*Suc* (*i* + *k*))) − *?fnik* + (*?fnik* − *?fni*) $\leq T_{max} + k * T_{max}$
        by (*rule add-mono* [*OF qp-suc-max*])
    thus *date* (*t A*•(*i* + *Suc k*)) − *date* (*t A*•*i*) ≤ (*Suc k*) $* T_{max}$
      by (*subst ntimes-suc-dist*) *simp*
  qed (*simp*)

lemma *qp-cone*:
  assumes *qp*:*quasiperiodic t*
  and $i \leq j$
  shows $(j - i) * T_{min} \leq$ *date* (*t A*•*j*) − *date* (*t A*•*i*)
        ∧ *date* (*t A*•*j*) − *date* (*t A*•*i*) $\leq (j - i) * T_{max}$
  proof −
    from ‹$i \leq j$› **have** $i + (j - i) = j$
      by *simp*
    from ‹$i \leq j$› **obtain** *k*
    where $k \geq 0$
    and $k = j - i$

by *simp*
from *qp* ‹$k \geq 0$› **have** $k * T_{min} \leq date\ (t\ A{\bullet}(i{+}k)) - date\ (t\ A{\bullet}i)$
by (*metis qp-cone-lower*)
with ‹$i + (j - i) = j$› **have** $(j - i) * T_{min} \leq date\ (t\ A{\bullet}j) - date\ (t\ A{\bullet}i)$
using ‹$k = j - i$› **by** *simp*
moreover from *qp* ‹$k \geq 0$› **have** $date\ (t\ A{\bullet}(i{+}k)) - date\ (t\ A{\bullet}i) \leq k * T_{max}$
by (*metis qp-cone-upper*)
with ‹$i + (j - i) = j$› **have** $date\ (t\ A{\bullet}j) - date\ (t\ A{\bullet}i) \leq (j{-}i) * T_{max}$
using ‹$k = j - i$› **by** *simp*
ultimately show *?thesis*
using ‹$i \leq j$› **by** (*simp add*:*real-of-nat-diff*)
qed

lemma *qp-date-ij*:
assumes *qp*: *quasiperiodic t*
and $i \neq j$
and $date\ (t\ A{\bullet}i) \leq date\ (t\ A{\bullet}j)$
shows $i < j$
proof (*rule ccontr*)
assume $\neg\ i < j$
hence $i \geq j$
by *simp*
hence $i > j$
using ‹$i \neq j$› **by** *simp*
hence $date\ (t\ A{\bullet}i) > date\ (t\ A{\bullet}j)$
proof −
from *qp* ‹$i{>}j$› **have** $date\ (t\ A{\bullet}j) + T_{min} \leq date\ (t\ A{\bullet}i)$
by (*rule qp-cone-lower-tmin*)
with *Tminpos* **show** *?thesis*
by *simp*
qed
thus *False*
using ‹$date\ (t\ A{\bullet}i) \leq date\ (t\ A{\bullet}j)$› **by** *auto*
qed

2.3 Happened before

lemma *hb1-reasonable*:
assumes *qp*:*quasiperiodic t*
and $e1 \mapsto t\ e2$
shows $date\ (t\ e1) < date(t\ e2)$
using *assms*(*2*) **proof** *induct*
fix $A\ B$ :: *node*
and $i\ j$ :: *nat*
assume $A = B$
and $i < j$
thus $date\ (t\ A{\bullet}i) < date(t\ B{\bullet}j)$
proof −
from *qp* ‹$i{<}j$› ‹$A = B$› **have** $date\ (t\ A{\bullet}i) + T_{min} \leq date\ (t\ B{\bullet}j)$
by (*simp add*: *qp-cone-lower-tmin*)
thus *?thesis*
using *Tminpos* **by** *simp*
qed

next
  fix *e1 e2*
  assume *arrival* (*t e1*) $\leq$ *date* (*t e2*)
  hence *date*(*t e1*) + *trans*(*t e1*) $\leq$ *date*(*t e2*)
    by *simp*
  moreover have τ_{min} $\leq$ *trans*(*t e1*)
    using *qp qp-trans* **by** *simp*
  ultimately show *date* (*t e1*) $<$ *date* (*t e2*)
    using *tauminpos* **by** *simp*
qed

lemma *hb-reasonable*:
assumes *qp*:*quasiperiodic t*
and *e1* $\rightarrow$*t e2*
shows *date* (*t e1*) $<$ *date*(*t e2*)
using *assms*(*2*) **proof** *induct*
  fix *y*
  assume *e1* $\mapsto$*t y*
  thus *date* (*t e1*) $<$ *date* (*t y*)
    using *qp* **by** (*simp add*: *hb1-reasonable*)
next
 fix *y z*
 assume *e1* $\rightarrow$*t y*
 and *y* $\mapsto$*t z*
 and *date* (*t e1*) $<$ *date* (*t y*)
 thus *date* (*t e1*) $<$ *date* (*t z*)
   using *qp* **by** (*metis dual-order.strict-trans hb1-reasonable*)
qed

lemma *hb-hb1-same*:
assumes *qp*:*quasiperiodic t*
and *A*•*i* $\rightarrow$*t A*•*j*
shows *A*•*i* $\mapsto$*t A*•*j*
proof −
from *qp* ⟨*A*•*i* $\rightarrow$*t A*•*j*⟩ **have** *date* (*t A*•*i*) $<$ *date* (*t A*•*j*)
  by (*rule hb-reasonable*)
with *qp* **have** *i* $<$ *j*
  by (*metis hb1-reasonable hb-subsequent less-asym linorder-neqE-nat*)
thus *?thesis*
  by (*simp add*: *hb-subsequent*)
qed

lemma *hb-A-ij*:
assumes *qp*:*quasiperiodic t*
and *A*•*i* $\rightarrow$*t A*•*j*
shows *i* $<$ *j*
proof (*rule ccontr*)
  assume *h*:$\neg$ *i* $<$ *j*
  from *assms* **have** *date* (*t A*•*i*) $<$ *date* (*t A*•*j*)
    by (*rule hb-reasonable*)
  show *False*
  proof (*cases i* = *j*)
    assume *i* = *j*

```
    hence date (t A•i) = date (t A•j)
      by simp
    thus False
      using ‹date (t A•i) < date (t A•j)› by simp
  next
    assume ¬ i = j
    with h have i > j by simp
    with qp have date (t A•i) > date (t A•j)
      by (simp add: hb1-reasonable hb-subsequent)
    thus False
      using ‹date (t A•i) < date (t A•j)› by simp
  qed
qed

lemma hb-node:
  assumes node x = node y
  and x ≠ y
  shows ¬ x ∥t y
  proof −
    from ‹node x = node y› obtain A i j
    where x = A•i
    and y = A•j
      by (metis (full-types) event.surjective unit.exhaust)
    with ‹x ≠ y› have i ≠ j
      by simp
    thus ?thesis
    proof (cases i < j)
      assume i < j
      with ‹x = A•i› ‹y = A•j› have x →t y
        by (simp add: hb-subsequent tranclp.r-into-trancl)
     thus ?thesis
       using concur-def by simp
    next
      assume ¬ i < j
      with ‹i ≠ j› have i > j
        by simp
      with  ‹x = A•i› ‹y = A•j› have y →t x
        by (simp add: hb-subsequent tranclp.r-into-trancl)
      thus ?thesis
        using concur-def by simp
    qed
  qed

lemma hb-concur-nodes:
  assumes (x →t y) ∧ (x ∥t z) ∧ (y ∥t z)
  shows node x ≠ node z ∧ node y ≠ node z
  proof (rule ccontr)
    assume h:¬ (node x ≠ node z ∧ node y ≠ node z)
    thus False
    proof (cases node x ≠ node z)
      assume node x ≠ node z
      with h have node y = node z
        by simp
```

```
      from assms have y ∥t z
        by simp
      show ?thesis
      proof (cases y = z)
        assume y = z
        moreover from assms have x →t y
          by simp
        ultimately show ?thesis
          using assms concur-def by simp
      next
        assume ¬ y = z
        with ‹node y = node z› have ¬ y ∥t z
          by (rule hb-node)
        with ‹y ∥t z› show ?thesis
          by simp
      qed
    next
      assume ¬ node x ≠ node z
      hence node x = node z
        by simp
      from assms have x ∥t z
        by simp
      show ?thesis
      proof (cases x = z)
        assume x = z
        moreover from assms have x →t y
          by simp
        ultimately show ?thesis
          using assms concur-def by simp
      next
        assume ¬ x = z
        with ‹node x = node z› have ¬ x ∥t z
          by (rule hb-node)
        with ‹x ∥t z› show ?thesis
          by simp
      qed
    qed
  qed


lemma hb-trans-arrival:
  assumes qp: quasiperiodic t
  and e1 →t e2
  shows node e1 = node e2 ∨
      (∃ e. node e = node e1
         ∧ date (t e) ≥ date (t e1)
         ∧ arrival (t e) ≤ date (t e2))
  using assms(2) proof (induct)
  case base
   fix y
   assume e1 ↦t y
     thus node e1 = node y ∨
          (∃e. node e = node e1
```

```
          ∧  date (t e1) ≤ date (t e)
          ∧ arrival (t e) ≤ date (t y))
   proof (cases node e1 = node y)
     assume node e1 = node y
     thus ?thesis
       by simp
   next
     assume ¬ node e1 = node y
     hence arrival (t e1) ≤ date (t y)
       using ‹e1 ↦t y› by (metis event.select-convs(1) hb1.cases)
     moreover have node e1 ≠ node y
       using ‹¬ node e1 = node y› by simp
     ultimately show ?thesis
       by auto
   qed
next
  case step
  fix y z
  assume e1 →t y
  and y ↦t z
  and e1y:node e1 = node y ∨
         (∃ e. node e = node e1
          ∧ date (t e1) ≤ date (t e)
          ∧ arrival (t e) ≤ date (t y))
  thus e1z:node e1 = node z ∨
          (∃ e. node e = node e1
           ∧ date (t e1) ≤ date (t e)
           ∧ arrival (t e) ≤ date (t z))
  proof (cases node e1 = node y)
    assume ¬ node e1 = node y
    then obtain e
    where arrival (t e) ≤ date (t y)
      using e1y by auto
    moreover have date (t y) < date (t z)
      using ‹y ↦t z› qp by (simp add: hb1-reasonable)
    ultimately show ?thesis
      using ‹node e1 ≠ node y› by (metis dual-order.trans e1y linear not-less)
  next
    assume node e1 = node y
    thus ?thesis
    proof (cases ¬ node z = node y)
      assume ¬ node z = node y
      hence arrival (t y) ≤ date (t z)
        using ‹y ↦t z› qp by (metis event.select-convs(1) hb1.cases)
      moreover have date (t e1) < date (t y)
        using ‹e1 →t y› qp by (simp add: hb-reasonable)
      ultimately show ?thesis
        using ‹node e1 = node y› by auto
    next
      assume ¬¬ node z = node y
      thus ?thesis
        using ‹node e1 = node y› by simp
    qed
```

qed
qed

lemma *hb-trans*:
assumes *qp*:*quasiperiodic t*
and $A{\cdot}i \rightarrow t\ B{\cdot}j$
and $\neg\ A{\cdot}i \mapsto t\ B{\cdot}j$
shows $\exists\ k{>}i.\ A{\cdot}k \mapsto t\ B{\cdot}j$
proof −
from *assms*(*2*−*3*) **have** $A \neq B$
by (*metis hb-hb1-same qp*)
then obtain *e*
where *node e* = *A*
and *date* (*t e*) ≥ *date* (*t A•i*)
and *arrival* (*t e*) ≤ *date* (*t B•j*)
using *assms* **by** (*metis event.select-convs*(*1*) *hb-trans-arrival*)
from ⟨*node e* = *A*⟩ **obtain** *k*
where *e* = *A•k*
by (*metis* (*full-types*) *event.surjective unit.exhaust*)
hence $A{\cdot}k \mapsto t\ B{\cdot}j$
using ⟨*arrival* (*t e*) ≤ *date* (*t B•j*)⟩ **by** (*simp add*: *hb-arrival*)
from *qp* ⟨$\neg\ A{\cdot}i \mapsto t\ B{\cdot}j$⟩ ⟨$A{\cdot}k \mapsto t\ B{\cdot}j$⟩ **have** $i \neq k$
by *auto*
moreover have *date* (*t A•k*) ≥ *date* (*t A•i*)
using ⟨*e* = *A•k*⟩ ⟨*date* (*t e*) ≥ *date* (*t A•i*)⟩ **by** *simp*
ultimately have $k{>}i$
using *qp* **by** (*simp add*: *qp-date-ij*)
thus *?thesis*
using ⟨$A{\cdot}k \mapsto t\ B{\cdot}j$⟩ **by** *metis*
qed

2.4 The set of predecessors

We prove here that the set of predecessors of an event with respect to the relation happened before is finite.

lemma *fin-Ai*:
assumes *qp*:*quasiperiodic t*
shows *finite* $\{A{\cdot}i \mid i.\ 0 \leq i \wedge date\ (t\ A{\cdot}i) < D\}$
proof −
from *qp* **have** $\forall i.\ i * T_{min} \leq date\ (t\ A{\cdot}i) - date\ (t\ A{\cdot}0)$
by (*metis monoid-add-class.add.left-neutral qp-cone-lower*)
moreover from *qp* **have** $date\ (t\ A{\cdot}0) \geq 0$
using *quasiperiodic-def* **by** *simp*
hence $\forall\ i.\ date\ (t\ A{\cdot}i) - date\ (t\ A{\cdot}0) \leq date\ (t\ A{\cdot}i)$
by *simp*
ultimately have $\forall i.\ i * T_{min} \leq date\ (t\ A{\cdot}i)$
by (*metis dual-order.trans*)
from *Tminpos* **obtain** *k* :: *nat*
where $k * T_{min} > D$
using *ntimes-bound* **by** *auto*
with ⟨$\forall i.\ i * T_{min} \leq date\ (t\ A{\cdot}i)$⟩ **have** $k * T_{min} \leq date\ (t\ A{\cdot}k)$

```
     by simp
   with ‹k * T_min > D› have date (t A•k) > D
     by simp
   hence {A•i | i. 0 ≤ i ∧ date (t A•i) < D} ⊆ {A•i | i. i < k}
   proof −
     assume D < date (t A•k)
     show {A•i | i. 0 ≤ i ∧ date (t A•i) < D} ⊆ {A•i | i. i < k}
     proof
       fix e
       assume e ∈ {A•i | i. 0 ≤ i ∧ date (t A•i) < D}
       then obtain j
       where 0 ≤ j
       and date (t A•j) < D
       and e = A•j
         by auto
       with ‹D < date (t A•k)› have date (t A•j) < date (t A•k)
         by simp
       hence date (t A•j) ≤ date (t A•k)
        by simp
       moreover have j ≠ k
       proof
         assume j = k
         hence j = k
           by simp
         hence date (t A•j) = date (t A•k)
           by simp
         thus False
           using ‹date (t A•j) < date (t A•k)› by simp
       qed
       ultimately have j < k
         using qp by (simp add: qp-date-ij)
       with ‹e = A•j› show e ∈ {A•i | i. i < k}
         by simp
     qed
   qed
   moreover have finite {A•i | i. i < k}
     by simp
   ultimately show ?thesis
     by (simp add: finite-subset)
  qed


lemma fin-node-date:
  assumes qp:quasiperiodic t
  shows finite {e. node e = A ∧ date (t e) < D}
  proof −
    have {e. node e = A ∧ date (t e) < D} ⊆ {A•i | i. 0 ≤ i ∧ date (t A•i) < D}
    proof
      fix e
      assume e ∈ {e. node e = A ∧ date (t e) < D}
      hence node e = A
      and date (t e) < D
        by auto
```

```
      then obtain j
      where 0 ≤ j
      and e = A•j
        by (metis (full-types) event.surjective le0 unit.exhaust)
      with ‹date (t e) < D› have date (t A•j) < D
        by simp
      with ‹0 ≤ j› ‹e = A•j› show e ∈ {A•i | i. 0 ≤ i ∧ date (t A•i) < D}
        by simp
    qed
    moreover have finite {A•i | i. 0 ≤ i ∧ date (t A•i) < D}
      using fin-Ai qp by simp
    ultimately show ?thesis
      by (metis (lifting, no-types) finite-subset)
  qed


lemma fin-date:
  assumes qp: quasiperiodic t
  shows finite {e. date (t e) < D}
  proof −
   from qp have ∀ A. finite {e. node e = A ∧ date (t e) < D}
     by (simp add: fin-node-date)
   with finnode have finite (⋃A ∈ nodes. {e. node e = A ∧ date (t e) < D})
     by simp
   moreover have (⋃A ∈ nodes. {e. node e = A ∧ date (t e) < D}) = {e. date (t e) < D}
   proof
     show {e. date (t e) < D} ⊆ (⋃A ∈ nodes. {e. node e = A ∧ date (t e) < D})
     proof
       fix e
       assume e ∈ {e. date (t e) < D}
       moreover from qp obtain A i
       where A ∈ nodes
       and e = A•i
         by (metis event.cases event.select-convs(1) node-coherent)
       ultimately have e ∈ {e. node e = A ∧ date (t e) < D}
         by auto
       thus e ∈ (⋃A ∈ nodes. {e. node e = A ∧ date (t e) < D})
         using ‹A ∈ nodes› by simp
     qed
     next
       show (⋃A ∈ nodes. {e. node e = A ∧ date (t e) < D}) ⊆ {e. date (t e) < D}
       proof
         fix e
         assume e ∈ (⋃A ∈ nodes. {e. node e = A ∧ date (t e) < D})
         hence date (t e) < D
           by simp
         moreover have e ∈ {e}
           by simp
         ultimately show e ∈ {e. date (t e) < D}
           by simp
       qed
   qed
   ultimately show ?thesis
```

by *simp*
qed

lemma *hb-finite*:
assumes *qp*: *quasiperiodic t*
shows *finite* $\{x.\ x \to t\ y\}$
proof −
from *qp* **have** $\forall x \in \{x.\ x \to t\ y\}.\ date\ (t\ x) < date\ (t\ y)$
by (*simp add*:*hb-reasonable*)
hence $\{x.\ x \to t\ y\} \subseteq \{x.\ date\ (t\ x) < date\ (t\ y)\}$
by *auto*
moreover with *qp* **have** *finite* $\{x.\ date\ (t\ x) < date\ (t\ y)\}$
by (*simp add*: *fin-date*)
ultimately show *?thesis*
by (*rule finite-subset*)
qed

lemma *hb-decr*:
assumes *qp*:*quasiperiodic t*
and $x \to t\ y$
shows $\{z.\ z \to t\ x\} \subset \{x.\ x {\to} t\ y\}$
proof
show $\{z.\ z \to t\ x\} \subseteq \{x.\ x {\to} t\ y\}$
proof
fix *z*
assume $z \in \{z.\ z \to t\ x\}$
with ‹$x \to t\ y$› **have** $z \to t\ y$
by *simp*
thus $z \in \{x.\ x {\to} t\ y\}$
by *simp*
qed
next
show $\{z.\ z \to t\ x\} \neq \{x.\ x {\to} t\ y\}$
proof −
have $\neg\ x \to t\ x$
proof
assume $x \to t\ x$
with *qp* **have** $date\ (t\ x) < date\ (t\ x)$
by (*rule hb-reasonable*)
thus *False*
by *simp*
qed
hence $x \notin \{z.\ z \to t\ x\}$
by *simp*
with ‹$x \to t\ y$› **show** $\{z.\ z \to t\ x\} \neq \{x.\ x {\to} t\ y\}$
by *auto*
qed
qed

3 Happened Before and Real-Time

The following lemmas link the relation happened before and the real-time dates of events.

lemma *not-hb-realtime*:
  assumes *quasiperiodic t*
  and $\neg(e1 \rightarrow t\ e2)$
  and *node e1* $\neq$ *node e2*
  shows *arrival* $(t\ e1) >$ *date* $(t\ e2)$
  using *assms* **by** (*metis hb-arrival leI tranclp.simps*)

lemma *hb-realtime-A*:
  assumes *qp*:*quasiperiodic t*
  and $A{\bullet}i \rightarrow t\ A{\bullet}j$
  shows $(j - i) * T_{min} \leq date\ (t\ A{\bullet}j) - date\ (t\ A{\bullet}i)$
    $\wedge\ date\ (t\ A{\bullet}j) - date\ (t\ A{\bullet}i) \leq (j - i) * T_{max}$
  proof −
    from *qp* $\langle A{\bullet}i \rightarrow t\ A{\bullet}j \rangle$ **have** $i < j$
      by (*metis hb-A-ij*)
    hence $i \leq j$
      by *simp*
    thus *?thesis*
      using *qp* **by** (*simp add*: *qp-cone*)
  qed

lemma *message-inversion*:
  assumes *qp*:*quasiperiodic t*
  and $T_{min} + \tau_{min} > \tau_{max}$
  shows $\forall\ C\ p\ q.\ p < q \longrightarrow arrival\ (t\ C{\bullet}p) < arrival\ (t\ C{\bullet}q)$
  proof (*rule ccontr*)
    assume $\neg\ (\forall\ C\ p\ q.\ p < q \longrightarrow arrival\ (t\ C{\bullet}p) < arrival\ (t\ C{\bullet}q))$
    then obtain *A i j*
    where $i < j$
    and $\neg(arrival\ (t\ A{\bullet}i) < arrival\ (t\ A{\bullet}j))$
      by *auto*
    hence $h{:}arrival\ (t\ A{\bullet}i) \geq arrival\ (t\ A{\bullet}j)$
      by *simp*
    from *qp* **obtain** *transmin*: $\tau_{min} \leq trans\ (t\ A{\bullet}j)$
    and *transmax*: $trans\ (t\ A{\bullet}i) \leq \tau_{max}$
      using *quasiperiodic-def* **by** *simp*
    from *transmin* **have** $arj{:}date\ (t\ A{\bullet}j) + \tau_{min} \leq arrival\ (t\ A{\bullet}j)$
      by *simp*
    from *transmax* **have** $ari{:}date\ (t\ A{\bullet}i) + \tau_{max} \geq arrival\ (t\ A{\bullet}i)$
      by *simp*
    from *qp* $\langle i < j \rangle$ **have** $subs{:}date\ (t\ A{\bullet}i) + T_{min} \leq date\ (t\ A{\bullet}j)$
      by (*rule qp-cone-lower-tmin*)
    from *arj h* **have** $date\ (t\ A{\bullet}j) + \tau_{min} \leq arrival\ (t\ A{\bullet}i)$
      by *simp*
    with *ari* **have** $date\ (t\ A{\bullet}j) + \tau_{min} \leq date\ (t\ A{\bullet}i) + \tau_{max}$
      by *simp*
    moreover from *subs arj* **have** $date\ (t\ A{\bullet}i) + T_{min} + \tau_{min} \leq date\ (t\ A{\bullet}j) + \tau_{min}$
      by *simp*
    ultimately have $T_{min} + \tau_{min} \leq \tau_{max}$

by *simp*
thus *False*
using *assms(2)* **by** *simp*
qed

lemma *hb-realtime*:
assumes *qp*: *quasiperiodic t*
and *mi*: $\forall$ *C p q. p* $<$ *q* $\longrightarrow$ *arrival* (*t C*•*p*) $<$ *arrival* (*t C*•*q*)
and $A \neq B$
shows *arrival* (*t A*•*i*) $\leq$ *date* (*t B*•*j*) $\longleftrightarrow$ *A*•*i* $\to$*t B*•*j*
proof
assume *A*•*i* $\to$*t B*•*j*
show *arrival* (*t A*•*i*) $\leq$ *date* (*t B*•*j*)
proof (*rule ccontr*)
assume *h*: $\neg$ *arrival* (*t A*•*i*) $\leq$ *date* (*t B*•*j*)
hence $\neg$ *A*•*i* $\mapsto$*t B*•*j*
using ‹$A \neq B$› **by** (*metis hb1-cases*)
then obtain *k*
where *i* $<$ *k*
and *A*•*k* $\mapsto$*t B*•*j*
using *qp* ‹$A \neq B$› ‹*A*•*i* $\to$*t B*•*j*› **by** (*metis hb-trans*)
hence *arrival* (*t A*•*k*) $\leq$ *date* (*t B*•*j*)
using ‹$A \neq B$› *qp* **by** (*metis hb1-cases*)
hence *arrival* (*t A*•*i*) $\geq$ *arrival* (*t A*•*k*)
using *h* **by** *simp*
with ‹*i*<*k*› **have** $\exists$ *C p q. p* $<$ *q* $\wedge$ *arrival* (*t C*•*p*) $\geq$ *arrival* (*t C*•*q*)
by *auto*
thus *False*
using *mi* **by** (*metis not-less*)
qed
next
assume *arrival* (*t A*•*i*) $\leq$ *date* (*t B*•*j*)
with *assms* **show** *A*•*i* $\to$*t B*•*j*
by (*metis hb-arrival tranclp.r-into-trancl*)
qed

4 Unitary Discretization

In this section we define the notion of unitary discretization and prove the central lemma, namely the existence of a unitary discretization is equivalent to a simple condition involving three events.

4.1 Discretization function

definition *discretization* :: (*event* $\Rightarrow$ *nat*) $\Rightarrow$ (*event* $\Rightarrow$ *tevent*) $\Rightarrow$ *bool*
where *discretization f t* = ($\forall$ *x y.* (*x* $\to$*t y*) $\longleftrightarrow$ *f x* $<$ *f y*)

function *discr* :: *trace* $\Rightarrow$ *event* $\Rightarrow$ *nat*
where *discr t y* = (*Max* (*insert 0* (*Suc* ‘ *discr t* ‘ {*x. x* $\to$*t y* $\wedge$ *quasiperiodic t*})))
by *auto*

```
termination
  proof (relation measure (λ(t, y). card {x. x →t y ∧ quasiperiodic t}))
    fix t y x
    assume x ∈ {x. x →t y ∧ quasiperiodic t}
    hence quasiperiodic t
    and x →t y
      by auto
    from hb-finite [OF this(1)] hb-decr [OF this]
    have card {z. z →t x} < card {x. x →t y}
      by (rule psubset-card-mono)
    thus ((t, x), (t, y)) ∈ measure (λ(t, y). card {x. x →t y ∧ quasiperiodic t})
      using ‹quasiperiodic t› by simp
  qed simp
declare discr.simps [simp del]

lemma qp-discr:
  assumes quasiperiodic t
  shows discr t y = (Max (insert 0 (Suc ‘ discr t ‘ {x. x →t y})))
  using assms by (simp add: discr.simps)

lemma case-discr [simp]:
  assumes qp: quasiperiodic t
  shows discr t y = (if {x. x →t y} = {} then 0
                     else Max ({discr t x | x. x →t y}) + 1)
  proof (cases {x. x →t y} = {})
    assume {x. x →t y} = {}
    moreover then have discr t y = 0
      using qp-discr [OF qp] by simp
    ultimately show ?thesis
      by simp
  next
    assume notempty: {x. x →t y} ≠ {}
    from hb-finite [OF qp] have finite (discr t ‘ {x. x →t y})
      by simp
    moreover have discr t ‘ {x. x →t y} = {discr t x|x. x →t y}
      by auto
    ultimately have finite {discr t x |x. x →t y}
      by simp
    from qp have discr t y = (Max (insert 0 (Suc ‘ discr t ‘ {x. x →t y})))
      by (rule qp-discr)
    also from ‹finite {discr t x |x. x →t y}› notempty
    have ... = Max (Suc ‘ {discr t x |x. x →t y})
      by (auto simp add: image-Collect)
    also from mono-Suc ‹finite {discr t x |x. x →t y}› notempty
    have ... = Suc (Max {discr t x |x. x →t y})
      by (subst mono-Max-commute [where f=Suc]) simp-all
    finally show ?thesis
      using notempty by simp
  qed
```

4.2 Central lemma UC is equivalent to UD

4.2.1 UD implies UC

```
lemma UD-concur:
  assumes UD:discretization f t
  and x ‖t y
  shows f x = f y
  proof (rule ccontr)
    assume ¬ f x = f y
    hence x →t y ∨ y →t x
    proof (cases f x < f y)
      assume f x < f y
      with UD show ?thesis
        using discretization-def by simp
    next
      assume ¬ f x < f y
      with ‹¬ f x = f y› have f x > f y
        by simp
      with UD show ?thesis
        using discretization-def by simp
    qed
    with ‹x ‖t y› show False
      using concur-def by simp
  qed

lemma UDUC:
  assumes UD:discretization f t
  shows UC:¬ (∃ x y z. (x →t y) ∧ (x ‖t z) ∧ (y ‖t z))
  proof
   assume (∃ x y z. (x →t y) ∧ (x ‖t z) ∧ (y ‖t z))
   then obtain x y z
   where x →t y
   and x ‖t z
   and y ‖t z
     by auto
   with UD have f x < f y
     using discretization-def by simp
   moreover from ‹x ‖t z› have f x = f z
     using assms by (simp add: UD-concur)
   moreover from ‹y ‖t z› have f y = f z
     using assms by (simp add: UD-concur)
   ultimately show False
     by simp
  qed
```

4.2.2 UC implies UD

```
lemma UCUD:
  assumes qp: quasiperiodic t
  and UC:¬ (∃ x y z. (x →t y) ∧ (x ‖t z) ∧ (y ‖t z))
  shows UD:∃ f. discretization f t
  proof −
    def f == λ x. discr t x
```

```
{ fix x y
  assume x →t y
  have f x < f y
  proof (rule ccontr)
    assume ¬ f x < f y
    with qp have finite {x. x→t y}
       by (simp add: hb-finite)
    hence finite {f x | x. x →t y}
      by simp
    from ‹x →t y› have f x ∈ {f x | x. x →t y}
      by auto
    with ‹finite {f x | x. x →t y}› have f x ≤ Max {f x::nat| x. x →t y}
      by simp
    hence f x < Max {f x::nat| x. x →t y} + 1
      by auto
    from ‹x →t y› have {x. x →t y} ≠ {}
      by auto
    with qp f-def have f y = Max {f x| x. x →t y} + 1
      by simp
    with ‹f x < Max {f x| x. x →t y} + 1› have  f x < f y
      by simp
    with ‹¬ f x < f y› show False
      by auto
  qed
}

moreover
{ fix x y
  assume f y < f x
  have f-hb1:¬ (x →t y)
  proof
    assume (x →t y)
    hence x →t y
      by auto
    hence f x < f y
      by (rule ‹⋀ya xa. xa →t ya ⟹ f xa < f ya›)
    with ‹f y < f x› show False
      by simp
  qed
}

moreover
{ fix x y z
  assume f y = f z + 1
  and f x < f y
  have ¬ z →t x
  proof
    assume z →t x
    hence f z < f x
      by (rule ‹⋀ya xa. xa →t ya ⟹ f xa < f ya›)
    moreover from ‹f y = f z + 1› ‹f x < f y› have f x ≤ f z
      by simp
```

```
    ultimately have f z ≥ f x
      by simp
    thus False
      using ‹f z < f x› by simp
  qed
}

moreover
{ fix y
  assume f y > 0
  have ∃ z. z →t y ∧ f y = f z + 1
   proof −
    have {x. x →t y} ≠ {}
    proof
      assume {x. x →t y} = {}
       hence {x. x →t y} = {}
        by simp
      with qp f-def have f y = 0
        by simp
      thus False
        using ‹f y > 0› by simp
    qed
    moreover with qp f-def have f y =  Max {f x| x. x →t y} + 1
      by simp
    moreover from qp have finite {x. x→t y}
      by (simp add: hb-finite)
    ultimately have ∃ x ∈ {x. x→t y}. f x = Max {f x |x. x ∈ {x. x→t y}}
      by (metis (mono-tags) finite-max)
    moreover have {x. x ∈ {x. x→t y}} = {x. x →t y}
      by simp
    ultimately obtain z
    where z ∈ {x. x→t y}
    and f z = Max{f x |x. x→t y}
      by auto
    with ‹f y =  Max {f x| x. x →t y} + 1› have f y = f z + 1
      by simp
    from ‹z ∈ {x. x→t y}› have z →t y
      by simp
    with ‹f y = f z + 1› show ?thesis
      by auto
  qed
}

moreover
{ fix x y
  assume f x < f y
  have x →t y
  proof (rule ccontr)
    assume ¬ x →t y
    from ‹f x < f y› have ¬ y →t x
      by (rule ‹⋀ya xa. f ya < f xa ⟹ ¬ xa →t ya›)
    with ‹¬ x →t y› have x ∥t y
      using concur-def by simp
```

from ‹$f\ x < f\ y$› **have** $f\ y > 0$
  by *simp*
with *f-def* **obtain** z
where $z \rightarrow t\ y$
and $f\ y = f\ z + 1$
  by (*metis* ‹$\bigwedge ya.\ 0 < f\ ya \Longrightarrow \exists z.\ \ z \rightarrow t\ ya \wedge f\ ya = f\ z + 1$›)
from ‹$\neg\ x \rightarrow t\ y$› ‹$z \rightarrow t\ y$› **have** $\neg\ x \rightarrow t\ z$
  by (*metis tranclp-trans*)
moreover from ‹$f\ y = f\ z + 1$› ‹$f\ x < f\ y$› **have** $\neg\ z \rightarrow t\ x$
  by (*rule* ‹$\bigwedge z\ ya\ xa.\ [\![f\ ya = f\ z + 1;\ f\ xa < f\ ya]\!] \Longrightarrow \neg\ z \rightarrow t\ xa$›)
ultimately have $z \parallel t\ x$
  using *concur-def* **by** *simp*
from ‹$z \rightarrow t\ y$› ‹$x \parallel t\ y$› ‹$z \parallel t\ x$› *UC* **show** *False*
  using *concur-def* **by** *auto*
qed
}

ultimately show *?thesis*
  using *discretization-def* **by** *metis*
qed

theorem *concur-discretization*:
  assumes *qp*:*quasiperiodic t*
  shows ($\exists f.$ *discretization f t*) $\longleftrightarrow$
    ($\neg$ ($\exists\ x\ y\ z.\ (x \rightarrow t\ y) \wedge (x \parallel t\ z) \wedge (y \parallel t\ z)$))
  proof
    assume $\exists f.$ *discretization f t*
    thus $\neg$ ($\exists\ x\ y\ z.\ (x \rightarrow t\ y) \wedge (x \parallel t\ z) \wedge (y \parallel t\ z)$)
      by (*metis UDUC*)
  next
    assume $\neg$ ($\exists\ x\ y\ z.\ (x \rightarrow t\ y) \wedge (x \parallel t\ z) \wedge (y \parallel t\ z)$)
    with *qp* **show** $\exists f.$ *discretization f t*
      by (*rule UCUD*)
  qed

4.3 Discretizing two nodes systems

In this section we give the weakest condition to ensure that a systems of two quasi-periodic nodes is unitary discretizable.

4.3.1 Soundness

lemma *discretization-2-sound*:
  assumes *qp*:*quasiperiodic t*
  and *card nodes* $= 2$
  and $T_{min} \geq 2 * \tau_{max}$
  shows $\neg$ ($\exists\ x\ y\ z.\ (x \rightarrow t\ y) \wedge (x \parallel t\ z) \wedge (y \parallel t\ z)$)
  proof
    assume ($\exists\ x\ y\ z.\ (x \rightarrow t\ y) \wedge (x \parallel t\ z) \wedge (y \parallel t\ z)$)
    then obtain $x\ y\ z$
    where $x \rightarrow t\ y$
    and $x \parallel t\ z$

and $y \parallel t\ z$
  by *auto*
with *qp* **have** *node* $x \neq$ *node* z **and** *node* $y \neq$ *node* z
  by (*simp-all add*: *hb-concur-nodes*)
with ‹*card nodes* = *2*› **have** *node* $x =$ *node* y
  by (*metis card2-xyz node-coherent*)
from ‹*node* $x =$ *node* y› ‹*node* $x \neq$ *node* z› **obtain** $A\ B\ i\ j\ k$
where $A \neq B$
and $x = A{\bullet}i$
and $y = A{\bullet}j$
and $z = B{\bullet}k$
  by (*metis* (*full-types*) *event.surjective unit.exhaust*)
with ‹$x \parallel t\ z$› ‹$y \parallel t\ z$› **have** h:($A{\bullet}i \parallel t\ B{\bullet}k$) $\wedge$ ($A{\bullet}j \parallel t\ B{\bullet}k$)
  by *simp*
from ‹$x \to t\ y$› ‹$x = A{\bullet}i$› ‹$y = A{\bullet}j$› **have** $A{\bullet}i \to t\ A{\bullet}j$
  by *simp*
with ‹$A{\bullet}i \to t\ A{\bullet}j$› *qp* **have** $i < j$
  by (*simp add*: *hb-A-ij*)
from *qp* **obtain** *transmin*: $\tau_{min} \leq$ *trans* ($t\ A{\bullet}j$)
and *transmax-A*: *trans* ($t\ A{\bullet}i$) $\leq \tau_{max}$
and *transmax-B*: *trans* ($t\ B{\bullet}k$) $\leq \tau_{max}$
  using *quasiperiodic-def* **by** *simp*
from *qp* ‹$i < j$› **have** *date* ($t\ A{\bullet}i$) + $T_{min} \leq$ *date* ($t\ A{\bullet}j$)
  by (*rule qp-cone-lower-tmin*)
moreover from h **have** $\neg$ ($A{\bullet}i \to t\ B{\bullet}k$)
  using *concur-def* **by** *simp*
hence *date* ($t\ B{\bullet}k$) < *arrival* ($t\ A{\bullet}i$)
  using *qp* ‹$A \neq B$› **by** (*metis event.select-convs*(*1*) *not-hb-realtime*)
hence *date* ($t\ B{\bullet}k$) < *date*($t\ A{\bullet}i$) + τ_{max}
  using *transmax-A* **by** *simp*
moreover from h **have** $\neg$ ($B{\bullet}k \to t\ A{\bullet}j$)
  using *concur-def* **by** *simp*
hence *date* ($t\ A{\bullet}j$) < *arrival* ($t\ B{\bullet}k$)
  using *qp* ‹$A \neq B$› **by** (*metis event.select-convs*(*1*) *not-hb-realtime*)
hence *date* ($t\ A{\bullet}j$) < *date* ($t\ B{\bullet}k$) + τ_{max}
  using *transmax-B* **by** *simp*
ultimately have *date* ($t\ A{\bullet}i$) + T_{min} < *date* ($t\ A{\bullet}i$) + τ_{max} + τ_{max}
  by *simp*
moreover have $\tau_{max} + \tau_{max} = 2 * \tau_{max}$
  by (*simp add*: *numerals*)
ultimately have *date* ($t\ A{\bullet}i$) + T_{min} < *date* ($t\ A{\bullet}i$) + $2 * \tau_{max}$
  by *simp*
thus *False*
  using ‹$2 * \tau_{max} \leq T_{min}$› **by** *simp*
qed

4.3.2 Weakest condition

lemma *discretization-2-eg*:
assumes *card nodes* = *2*
and $T_{min} < 2 * \tau_{max}$
shows $\exists\, t$. (*quasiperiodic* t) $\wedge$ ($\exists\ x\ y\ z$. ($x \to t\ y$) $\wedge$ ($x \parallel t\ z$) $\wedge$ ($y \parallel t\ z$))
proof −

from ‹*card nodes* = *2*› *finnode* **obtain** *A*
where $A \in nodes$
  using *node-coherent* **by** *auto*
with ‹*card nodes* = *2*› *finnode* **have** $card\ (nodes - \{A\}) = 1$
  by *simp*
then obtain *B*
where $B \in nodes - \{A\}$
  by (*metis card-empty equals0I zero-neq-one*)
from ‹$A \in nodes$› ‹$B \in nodes - \{A\}$› **have** *abnodes*: $A \in nodes\ B \in nodes\ A \neq B$
  by *auto*
from ‹$T_{min} < 2 * \tau_{max}$› **obtain** ε
where $\varepsilon/2 > 0$
and $\varepsilon = 2 * \tau_{max} - T_{min}$
  by *simp*
hence $\tau_{max} - \varepsilon/2 = T_{min}/2$
  by *linarith*
moreover with *Tminpos* **have** $\tau_{max} - \varepsilon/2 > 0$
  by *simp*
ultimately have $\tau_{max} - \varepsilon/2 < \tau_{max}$
  using *taumaxpos* ‹$0 < \varepsilon\ /\ 2$› **by** *simp*

def *eg* == $\lambda e :: event.$
  if node e = *A then* ⦇ $date = act\ e * T_{min},\ trans = \tau_{max}$ ⦈
  else if node e = *B then* ⦇ $date = \tau_{max} - \varepsilon/2 + act\ e * T_{min},\ trans = \tau_{max}$ ⦈
  else ⦇ $date = act\ e * T_{min},\ trans = \tau_{max}$ ⦈

have $\forall\ e.\ trans\ (eg\ e) = \tau_{max}$
  using *eg-def* **by** *simp*
hence $\forall\ e.\ \tau_{min} \leq trans\ (eg\ e) \wedge trans\ (eg\ e) \leq \tau_{max}$
  using *eg-def taubounds* **by** *simp*
moreover have $\forall\ e.\ date\ (eg\ (step\ e)) - date\ (eg\ e) = T_{min}$
  using *eg-def* **by** *simp*
hence $\forall\ e.\ T_{min} \leq date\ (eg\ (step\ e)) - date\ (eg\ e)$
  $\wedge\ date\ (eg\ (step\ e)) - date\ (eg\ e) \leq T_{max}$
  using *Tbounds* **by** *simp*
moreover have $\forall\ e.\ 0 \leq date(eg\ e)$
  using *eg-def Tminpos* ‹$\tau_{max} - \varepsilon/2 > 0$› **by** *simp*
ultimately have *qp*:*quasiperiodic eg*
  using *quasiperiodic-def* **by** *simp*

have *mi*:$\forall\ N\ p\ q.\ p < q \longrightarrow arrival\ (eg\ N{\cdot}p) < arrival\ (eg\ N{\cdot}q)$
  using *eg-def Tminpos* **by** (*simp add*: ‹$\forall e.\ tevent.trans\ (eg\ e) = \tau_{max}$›)

have *a0*:$eg\ A{\cdot}0$ = ⦇ $date = 0,\ trans = \tau_{max}$ ⦈
and *b0*:$eg\ B{\cdot}0$ = ⦇ $date = \tau_{max} - \varepsilon/2,\ trans = \tau_{max}$ ⦈
and *a1*:$eg\ A{\cdot}1$ = ⦇ $date = T_{min},\ trans = \tau_{max}$ ⦈
  using *abnodes eg-def* **by** *simp-all*

have $A{\cdot}0 \rightarrow eg\ A{\cdot}1 \wedge A{\cdot}0\ \|eg\ B{\cdot}0 \wedge B{\cdot}0\ \|eg\ A{\cdot}1$
proof −
  have $A{\cdot}0 \rightarrow eg\ A{\cdot}1$
    by (*simp add*: *hb-subsequent tranclp.r-into-trancl*)

moreover
{ from *a0 b0* ‹$\tau_{max} - \varepsilon/2 < \tau_{max}$› **have** *date* (*eg B•0*) < *arrival* (*eg A•0*)
by *simp*
with *qp mi* ‹$A \neq B$› **have** ¬ *A•0* →*eg B•0*
by (*metis hb-realtime not-le*)
moreover from ‹$\tau_{max} - \varepsilon/2 > 0$› *taumaxpos this a0 b0*
have *date* (*eg A•0*) < *arrival* (*eg B•0*)
by *simp*
hence ¬ *B•0* →*eg A•0*
using ‹$A \neq B$› *mi qp* **by** (*metis hb-realtime not-le*)
ultimately have *A•0* ‖*eg B•0*
using *concur-def* **by** *simp*
}

moreover
{ from *b0 a1 Tmaxpos* ‹$\tau_{max} - \varepsilon/2 = T_{min}/2$› *Tminpos* **have** *date* (*eg B•0*) < *date* (*eg A•1*)
by *simp*
moreover with *a1 taumaxpos* **have** *arrival*(*eg A•1*) > *date*(*eg A•1*)
by *auto*
ultimately have *date* (*eg B•0*) < *arrival* (*eg A•1*)
by *simp*
hence ¬ *A•1* →*eg B•0*
using ‹$A \neq B$› *mi qp* **by** (*metis hb-realtime not-le*)
moreover from *b0* ‹$\tau_{max} - \varepsilon\ /\ 2 = T_{min}\ /\ 2$› **have** *arrival* (*eg B•0*) = $\tau_{max} + T_{min}/2$
by *simp*
hence *date* (*eg A•1*) < *arrival* (*eg B•0*)
using *a1* ‹$\tau_{max} - \varepsilon\ /\ 2 < \tau_{max}$› ‹$\tau_{max} - \varepsilon\ /\ 2 = T_{min}\ /\ 2$› **by** *simp*
hence ¬ *B•0* →*eg A•1*
using ‹$A \neq B$› *mi qp* **by** (*metis hb-realtime not-le*)
ultimately have *b0a1*:*B•0* ‖*eg A•1*
using *concur-def* **by** *simp*
}

ultimately show *?thesis*
by *simp*
qed
with *qp* **show** *?thesis*
using *concur-def* **by** *auto*
qed

lemma *discretization-2-weakest*:
assumes *card nodes* = *2*
and ∀ *t*. (*quasiperiodic t*) ⟶ ¬(∃ *x y z*. (*x* →*t y*) ∧ (*x* ‖*t z*) ∧ (*y* ‖*t z*))
shows $T_{min} \geq 2 * \tau_{max}$
proof (*rule ccontr*)
assume ¬ $T_{min} \geq 2 * \tau_{max}$
hence $T_{min} < 2 * \tau_{max}$
by *simp*
with ‹*card nodes* = *2*›
have ∃ *t*. (*quasiperiodic t*) ∧ (∃ *x y z*. (*x* →*t y*) ∧ (*x* ‖*t z*) ∧ (*y* ‖*t z*))
by (*rule discretization-2-eg*)
thus *False*
using *assms*(*2*) **by** *auto*

qed

theorem *discretization-2*:
assumes *card nodes* = *2*
and *quasiperiodic t*
shows ($\forall t.$ *quasiperiodic t* $\longrightarrow$ ($\exists f.$ *discretization f t*)) $\longleftrightarrow$ ($T_{min} \geq 2 * \tau_{max}$)
proof −
have $\forall t.$ *quasiperiodic t* $\longrightarrow$
($\exists f.$ *discretization f t*) $\longleftrightarrow$ $\neg$ ($\exists$ *x y z*. ($x \rightarrow t\ y$) $\wedge$ ($x \parallel t\ z$) $\wedge$ ($y \parallel t\ z$))
by (*simp add*: *concur-discretization*)
moreover have ($\forall t.$ *quasiperiodic t* $\longrightarrow$ $\neg$ ($\exists$ *x y z*. ($x \rightarrow t\ y$) $\wedge$ ($x \parallel t\ z$) $\wedge$ ($y \parallel t\ z$)))
$\longleftrightarrow$ ($T_{min} \geq 2 * \tau_{max}$)
proof
assume $\forall t.$ *quasiperiodic t* $\longrightarrow$ $\neg$ ($\exists$ *x y z*. ($x \rightarrow t\ y$) $\wedge$ ($x \parallel t\ z$) $\wedge$ ($y \parallel t\ z$))
with ‹*card nodes* = *2*› **show** $T_{min} \geq 2 * \tau_{max}$
by (*rule discretization-2-weakest*)
next
assume $T_{min} \geq 2 * \tau_{max}$
show $\forall t.$ *quasiperiodic t* $\longrightarrow$ $\neg$ ($\exists$ *x y z*. ($x \rightarrow t\ y$) $\wedge$ ($x \parallel t\ z$) $\wedge$ ($y \parallel t\ z$))
proof
fix *t*
show *quasiperiodic t* $\longrightarrow$ $\neg$ ($\exists$ *x y z*. ($x \rightarrow t\ y$) $\wedge$ ($x \parallel t\ z$) $\wedge$ ($y \parallel t\ z$))
proof
assume *qp*:*quasiperiodic t*
thus $\neg$ ($\exists$ *x y z*. ($x \rightarrow t\ y$) $\wedge$ ($x \parallel t\ z$) $\wedge$ ($y \parallel t\ z$))
using ‹$T_{min} \geq 2 * \tau_{max}$› ‹*card nodes* = *2*› **by** (*metis discretization-2-sound*)
qed
qed
qed
ultimately show *?thesis*
by *simp*
qed

4.4 Discretizing general systems

We prove here that general quasi-periodic systems of more than two nodes are not unitary discretizable.

lemma *discretization-eg*:
assumes *2* < *card nodes*
shows $\exists$ *t x y z*. *quasiperiodic t* $\wedge$ $x \rightarrow t\ y \wedge x \parallel t\ z \wedge y \parallel t\ z$
proof −
from *assms*(*1*) *finnode* **obtain** *A*
where $A \in$ *nodes*
by (*metis all-not-in-conv card-eq-0-iff less-nat-zero-code*)
moreover with *assms finnode* **obtain** *B*
where $B \in$ *nodes* − {*A*}
by (*metis Suc-1 Suc-leI card-0-eq card-Suc-Diff1 finite.cases*
finite-Diff insertI1 less-le-trans not-less-iff-gr-or-eq zero-less-Suc)
ultimately obtain *C*
where $C \in$ *nodes* − {*A*, *B*}
using *assms finnode* **by** (*metis Diff-insert2 One-nat-def Suc-1 Suc-leI*
all-not-in-conv card.insert-remove card-empty finite.emptyI insert-Diff-single

insert-absorb lessI not-le)
from ‹$A \in nodes$› ‹$B \in nodes - \{A\}$› ‹$C \in nodes - \{A, B\}$›
have *abcnodes*: $A \in nodes$ $B \in nodes$ $C \in nodes$ $A \neq B$ $B \neq C$ $C \neq A$
by *auto*

def $eg == \lambda e :: event.$
$\quad$ *if node e = A then* ⦇ $date = act\ e * T_{max},\ trans = \tau_{max}$ ⦈
$\quad$ *else if node e = B then* ⦇ $date = \tau_{max} + act\ e * T_{max},\ trans = \tau_{max}$ ⦈
$\quad$ *else if node e = C then* ⦇ $date = \tau_{max}\ /\ 2 + act\ e * T_{max},\ trans = \tau_{max}$ ⦈
$\quad$ *else* ⦇ $date = act\ e * T_{max},\ trans = \tau_{max}$ ⦈

have $\forall\ e.\ trans\ (eg\ e) = \tau_{max}$
using *eg-def* **by** *simp*
hence $\forall\ e.\ \tau_{min} \leq trans\ (eg\ e) \wedge trans\ (eg\ e) \leq \tau_{max}$
using *eg-def taubounds* **by** *simp*
moreover have $\forall\ e.\ date\ (eg\ (step\ e)) - date\ (eg\ e) = T_{max}$
using *eg-def* **by** *simp*
hence $\forall\ e.\ T_{min} \leq date\ (eg\ (step\ e)) - date\ (eg\ e)$
$\wedge\ date\ (eg\ (step\ e)) - date\ (eg\ e) \leq T_{max}$
using *Tbounds* **by** *simp*
moreover have $\forall\ e.\ 0 \leq date(eg\ e)$
using *eg-def Tmaxpos taumaxpos'* **by** *simp*
ultimately have *qp*:*quasiperiodic eg*
using *quasiperiodic-def* **by** *simp*

have *mi*:$\forall\ N\ p\ q.\ p < q \longrightarrow arrival\ (eg\ N{\cdot}p) < arrival\ (eg\ N{\cdot}q)$
using *eg-def Tmaxpos* **by** (*simp add*: ‹$\forall e.\ tevent.trans\ (eg\ e) = \tau_{max}$›)

from *abcnodes*
have *a0*:$eg\ A{\cdot}0 =$ ⦇ $date = 0,\ trans = \tau_{max}$ ⦈
and *b0*:$eg\ B{\cdot}0 =$ ⦇ $date = \tau_{max},\ trans = \tau_{max}$ ⦈
and *c0*:$eg\ C{\cdot}0 =$ ⦇ $date = \tau_{max}\ /\ 2,\ trans = \tau_{max}$ ⦈
using *eg-def* **by** *simp-all*

have $A{\cdot}0 \rightarrow eg\ B{\cdot}0 \wedge A{\cdot}0 \parallel eg\ C{\cdot}0 \wedge B{\cdot}0 \parallel eg\ C{\cdot}0$
proof −
{ from *a0 b0* **have** $arrival\ (eg\ A{\cdot}0) \leq date\ (eg\ B{\cdot}0)$
by *simp*
hence $A{\cdot}0 \rightarrow eg\ B{\cdot}0$
using *qp a0 b0* ‹$A \neq B$› *not-hb-realtime* **by** *fastforce*
}

moreover
{ from *taumaxpos* **have** $\tau_{max}/2 < \tau_{max}$
by *simp*
from *this a0 c0 tauminpos* **have** $date\ (eg\ C{\cdot}0) < arrival\ (eg\ A{\cdot}0)$
by *simp*
with *qp mi* ‹$C \neq A$› **have** $\neg\ A{\cdot}0 \rightarrow eg\ C{\cdot}0$
by (*metis hb-realtime not-le*)
moreover from *taumaxpos* **have** $0 < \tau_{max}/2 + \tau_{max}$
by *simp*
from *this a0 c0* **have** $date\ (eg\ A{\cdot}0) < arrival\ (eg\ C{\cdot}0)$
by *simp*

```
      hence ¬ C·0 →eg A·0
        using ‹C ≠ A› mi qp by (metis hb-realtime not-le)
      ultimately have A·0 ∥eg C·0
        using concur-def by simp
    }

    moreover
    { from taumaxpos have τmax/2 < τmax + τmax
        by simp
      from this b0 c0 tauminpos have date (eg C·0) < arrival (eg B·0)
        by simp
      hence ¬ B·0 →eg C·0
        using ‹B ≠ C› mi qp by (metis hb-realtime not-le)
      moreover from taumaxpos have τmax < τmax/2 + τmax
        by simp
      with b0 c0 have date (eg B·0) < arrival (eg C·0)
        by simp
      hence ¬ C·0 →eg B·0
        using ‹B ≠ C› mi qp by (metis hb-realtime not-le)
      ultimately have B·0 ∥eg C·0
        using concur-def by simp
    }

    ultimately show ?thesis
      by simp
  qed
  with qp show ?thesis
    by auto
qed

theorem discretization:
  assumes 2 < card nodes
  shows ∃ t. quasiperiodic t ∧ ¬ (∃ f. discretization f t)
  proof −
    have (∃ t. quasiperiodic t ∧ ¬ (∃ f. discretization f t))
          ⟷ (∃ t x y z. quasiperiodic t ∧ (x →t y) ∧ (x ∥t z) ∧ (y ∥t z))
    by (metis concur-discretization)
    with ‹2 < card nodes› show ?thesis
      by (metis discretization-eg)
  qed
```

5 Quasi-synchronous Abstraction

In this section we link the quasi-synchronous abstraction of Caspi with quasi-periodic system of two nodes.

5.1 Soundness

```
lemma not-quasi-synchrony-sound-case1:
  assumes qp:quasiperiodic t
  and A ≠ B
```

and $T_{min} \geq 2 * \tau_{max}$
and $n > 0$
and $(\neg\ A{\cdot}i \rightarrow t\ B{\cdot}j)$
and $(A{\cdot}(i{+}n) \rightarrow t\ B{\cdot}(j{+}1))$
shows $n * T_{min} + \tau_{min} < T_{max} + \tau_{max}$
proof −
from ‹$T_{min} \geq 2 * \tau_{max}$› *taumaxpos' tauminpos* **have** $T_{min} + \tau_{min} > \tau_{max}$
by *simp*
with *qp* **have** *mi*:$\forall\ C\ p\ q.\ p < q \longrightarrow arrival\ (t\ C{\cdot}p) < arrival\ (t\ C{\cdot}q)$
by (*rule message-inversion*)
from *qp* **obtain** *transmin*: $\tau_{min} \leq trans\ (t\ A{\cdot}(i{+}n))$
and *transmax*: $trans\ (t\ A{\cdot}i) \leq \tau_{max}$
using *quasiperiodic-def* **by** *auto*
from *qp* ‹$A \neq B$› ‹$\neg\ A{\cdot}i \rightarrow t\ B{\cdot}j$› **have** $arrival\ (t\ A{\cdot}i) > date\ (t\ B{\cdot}j)$
by (*metis event.select-convs(1) not-hb-realtime*)
hence $date\ (t\ B{\cdot}j) < date\ (t\ A{\cdot}i) + \tau_{max}$
using *transmax* **by** *simp*
moreover from *qp* ‹$A \neq B$› *mi* ‹$A{\cdot}(i{+}n) \rightarrow t\ B{\cdot}(j{+}1)$›
have $date\ (t\ B{\cdot}(j{+}1)) \geq arrival\ (t\ A{\cdot}(i{+}n))$
by (*metis hb-realtime*)
hence $date\ (t\ B{\cdot}(j{+}1)) \geq date\ (t\ A{\cdot}(i{+}n)) + \tau_{min}$
using *transmin* **by** *simp*
moreover from *qp* **have** $date\ (t\ B{\cdot}(j{+}1)) - date\ (t\ B{\cdot}j) \leq T_{max}$
by (*metis Suc-eq-plus1 qp-suc*)
hence $date\ (t\ B{\cdot}(j{+}1)) \leq date\ (t\ B{\cdot}j) + T_{max}$
by *simp*
moreover have $n * T_{min} \leq date\ (t\ A{\cdot}(i{+}n)) - date\ (t\ A{\cdot}i)$
using *qp* ‹$n > 0$› **by** (*simp add*: *qp-cone-lower*)
ultimately show $n * T_{min} + \tau_{min} < T_{max} + \tau_{max}$
by *simp*
qed

lemma *not-quasi-synchrony-sound-case2*:
assumes *qp*:*quasiperiodic t*
and $A \neq B$
and $T_{min} \geq 2 * \tau_{max}$
and $n > 0$
and $(\neg\ A{\cdot}i \rightarrow t\ B{\cdot}j)$
and $(\neg\ B{\cdot}(j{+}1) \rightarrow t\ A{\cdot}(i{+}n))$
shows $n * T_{min} < T_{max} + 2 * \tau_{max}$
proof −
from ‹$T_{min} \geq 2 * \tau_{max}$› *tauminpos taumaxpos'* **have** *mi*:$T_{min} + \tau_{min} > \tau_{max}$
by *simp*
from *qp* **obtain** *transmaxb*: $trans\ (t\ B{\cdot}(j{+}1)) \leq \tau_{max}$
and *transmaxa*: $trans\ (t\ A{\cdot}i) \leq \tau_{max}$
using *quasiperiodic-def* **by** *auto*
from *qp* ‹$A \neq B$› ‹$\neg\ A{\cdot}i \rightarrow t\ B{\cdot}j$› **have** $arrival\ (t\ A{\cdot}i) > date\ (t\ B{\cdot}j)$
by (*metis event.select-convs(1) not-hb-realtime*)
hence $date\ (t\ B{\cdot}j) < date\ (t\ A{\cdot}i) + \tau_{max}$
using *transmaxa* **by** *simp*
moreover from *qp* ‹$A \neq B$› ‹$\neg\ B{\cdot}(j{+}1) \rightarrow t\ A{\cdot}(i{+}n)$›
have $date\ (t\ A{\cdot}(i{+}n)) < arrival\ (t\ B{\cdot}(j{+}1))$

by (*metis event.select-convs*(*1*) *not-hb-realtime*)
hence $date\ (t\ A{\bullet}(i{+}n)) < date\ (t\ B{\bullet}(j{+}1)) + \tau_{max}$
using *transmaxb* **by** *simp*
moreover from *qp* **have** $date\ (t\ B{\bullet}(j{+}1)) - date\ (t\ B{\bullet}j) \leq T_{max}$
by (*metis Suc-eq-plus1 qp-suc*)
hence $date\ (t\ B{\bullet}(j{+}1)) \leq date\ (t\ B{\bullet}j) + T_{max}$
by *simp*
moreover have $n * T_{min} \leq date\ (t\ A{\bullet}(i{+}n)) - date\ (t\ A{\bullet}i)$
using *qp* ‹$n > 0$› **by** (*simp add*: *qp-cone-lower*)
ultimately show $n * T_{min} < T_{max} + 2{*}\tau_{max}$
by *simp*
qed

lemma *quasi-synchrony-sound*:
assumes *discretization f t*
and *qp*:*quasiperiodic t*
and $A \neq B$
and $T_{min} \geq 2 * \tau_{max}$
and $n > 0$
and $n * T_{min} \geq T_{max} + 2 * \tau_{max}$
shows $\neg\ (f\ B{\bullet}j \leq f\ A{\bullet}i \wedge f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1))$
proof
assume $(f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1))$
hence $f\ B{\bullet}j \leq f\ A{\bullet}i$
and $f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1)$
by *simp-all*
moreover from ‹$n * T_{min} \geq T_{max} + 2 * \tau_{max}$› *taumaxpos′ tauminpos′*
have $n * T_{min} + \tau_{min} \geq T_{max} + \tau_{max}$
by *simp*
ultimately show *False*
proof (*cases* $f\ A{\bullet}(i{+}n) = f\ B{\bullet}(j{+}1)$)
assume $f\ A{\bullet}(i{+}n) = f\ B{\bullet}(j{+}1)$
with *assms*(*1*) **have** $\neg\ B{\bullet}(j{+}1) \to t\ A{\bullet}(i{+}n)$
using *discretization-def* **by** *simp*
moreover from *assms*(*1*) ‹$f\ B{\bullet}j \leq f\ A{\bullet}i$› **have** $\neg\ A{\bullet}i \to t\ B{\bullet}j$
using *discretization-def* **by** *simp*
ultimately have $n * T_{min} < T_{max} + 2{*}\tau_{max}$
using *assms*(*2*−*5*) **by** (*simp add*: *not-quasi-synchrony-sound-case2*)
thus *False*
using ‹$n * T_{min} \geq T_{max} + 2 * \tau_{max}$› **by** *simp*
next
assume $\neg\ f\ A{\bullet}(i{+}n) = f\ B{\bullet}(j{+}1)$
with ‹$f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1)$› **have** $f\ A{\bullet}(i{+}n) < f\ B{\bullet}(j{+}1)$
by *simp*
with *assms*(*1*) **have** $A{\bullet}(i{+}n) \to t\ B{\bullet}(j{+}1)$
using *discretization-def* **by** *simp*
moreover from *assms*(*1*) ‹$f\ B{\bullet}j \leq f\ A{\bullet}i$› **have** $\neg\ A{\bullet}i \to t\ B{\bullet}j$
using *discretization-def* **by** *simp*
ultimately have $n * T_{min} + \tau_{min} < T_{max} + \tau_{max}$
using *assms*(*2*−*5*) **by** (*simp add*: *not-quasi-synchrony-sound-case1*)
thus *False*
using ‹$n * T_{min} + \tau_{min} \geq T_{max} + \tau_{max}$› **by** *simp*

qed
qed

5.2 Weakest condition

lemma *quasi-synchrony-eg*:
assumes $A \neq B$
and $T_{min} \geq 2 * \tau_{max}$
and $n > 0$
and $\neg\ (\exists\ t\ i\ j.\ quasiperiodic\ t$
$\wedge\ (\neg\ A{\bullet}i \rightarrow t\ B{\bullet}j) \wedge (\neg\ B{\bullet}(j{+}1) \rightarrow t\ A{\bullet}(i{+}n)))$
shows $n * T_{min} \geq T_{max} + 2 * \tau_{max}$
proof (*rule ccontr*)
assume $\neg\ n * T_{min} \geq T_{max} + 2{*}\tau_{max}$
hence $\exists\ t\ i\ j.\ quasiperiodic\ t$
$\wedge\ (\neg\ A{\bullet}i \rightarrow t\ B{\bullet}j) \wedge (\neg\ B{\bullet}(j{+}1) \rightarrow t\ A{\bullet}(i{+}n))$
proof −
from $\langle T_{min} \geq 2 * \tau_{max}\rangle$ *tauminpos taumaxpos* **have** $mit{:}T_{min} + \tau_{min} > \tau_{max}$
by *simp*
from $\langle\neg\ n * T_{min} \geq T_{max} + 2{*}\tau_{max}\rangle$ **obtain** ε
where $\varepsilon/2 > 0$
and $\varepsilon = T_{max} + 2 * \tau_{max} - n * T_{min}$
by *simp*

def $eg == \lambda e :: event.$
$if\ node\ e = A \wedge act\ e = 0\ then\ (|\ date = \varepsilon/2,\ trans = \tau_{max}\ |)$
$else\ if\ node\ e = B\ then\ (|\ date = \tau_{max} + act\ e * T_{max},\ trans = \tau_{max}\ |)$
$else\ (|\ date = \varepsilon/2 + act\ e * T_{min},\ trans = \tau_{max}\ |)$

have $\forall\ e.\ trans\ (eg\ e) = \tau_{max}$
using *eg-def* **by** *simp*
hence $\forall\ e.\ \tau_{min} \leq trans\ (eg\ e) \wedge trans\ (eg\ e) \leq \tau_{max}$
using *eg-def taubounds* **by** *simp*
moreover have $\forall\ e.\ date\ (eg\ (step\ e)) - date\ (eg\ e) = T_{min}$
$\vee\ date\ (eg\ (step\ e)) - date\ (eg\ e) = T_{max}$
using *eg-def* $\langle A \neq B\rangle$ **by** *simp*
hence $\forall\ e.\ T_{min} \leq date\ (eg\ (step\ e)) - date\ (eg\ e)$
$\wedge\ date\ (eg\ (step\ e)) - date\ (eg\ e) \leq T_{max}$
using *Tbounds* **by** *auto*
moreover have $\forall\ e.\ 0 \leq date(eg\ e)$
using *eg-def Tmaxpos Tminpos taumaxpos'* $\langle\varepsilon/2 > 0\rangle$ **by** *simp*
ultimately have $qp{:}quasiperiodic\ eg$
using *quasiperiodic-def* **by** *simp*

have $a0{:}arrival\ (eg\ A{\bullet}0) = \tau_{max} + \varepsilon/2$
and $b0{:}date\ (eg\ B{\bullet}0) = \tau_{max}$
and $b1{:}arrival\ (eg\ B{\bullet}1) =\ T_{max} + 2{*}\tau_{max}$
and $an{:}date\ (eg\ A{\bullet}n) = \varepsilon/2 + n * T_{min}$
using *eg-def* $\langle A \neq B\rangle$ **by** *simp-all*

from *qp mit* **have** $mi{:}\forall\ C\ p\ q.\ p < q \longrightarrow arrival\ (eg\ C{\bullet}p) < arrival\ (eg\ C{\bullet}q)$
by (*rule message-inversion*)

```
      from ‹ε = T_max + 2*τ_max − n * T_min› ‹ε/2 > 0› have ε/2 + n * T_min < T_max + 2*τ_max
        by linarith
      with an b1  have date (eg A·n) < arrival (eg B·1)
        by simp
      with qp mi ‹A ≠ B› have ¬ B·1 →eg A·n
        by (metis hb-realtime not-le)
      moreover from a0 b0 ‹ε/2 >0› taumaxpos have date (eg B·0) < arrival (eg A·0)
        by auto
      with qp mi ‹A ≠ B› have ¬ A·0 →eg B·0
        by (metis hb-realtime not-le)
      ultimately have (¬ A·0 →eg B·0) ∧ (¬ B·1 →eg A·n)
        by simp
      thus ?thesis
        using qp by fastforce
    qed
    with assms(4) show False
      by auto
  qed


lemma quasi-synchrony-weakest:
  assumes card nodes = 2
  and A ≠ B
  and T_min ≥ 2 * τ_max
  and n > 0
  and ¬ (∃ t i j f.
          quasiperiodic t ∧
          discretization f t ∧
          (f B·j ≤ f A·i ∧  f A·(i+n) ≤ f B·(j+1)))
  shows n * T_min ≥ T_max + 2*τ_max
  proof (rule ccontr)
    assume ¬ n * T_min ≥ T_max + 2*τ_max
    with assms(2−4)
    have ∃ t i j f.
           quasiperiodic t ∧
           (¬ A·i →t B·j) ∧ (¬ B·(j+1) →t A·(i+n))
      by (metis quasi-synchrony-eg)
    then obtain t i j
    where qp:quasiperiodic t
    and (¬ A·i →t B·j)
    and (¬ B·(j+1) →t A·(i+n))
      by auto
    moreover from assms(1) assms(3) qp obtain f
    where f-discr:discretization f t
      by (metis discretization-2)
    moreover have (f B·j ≤ f A·i ∧  f A·(i+n) ≤ f B·(j+1))
    proof −
      from f-discr ‹(¬ A·i →t B·j)› have f B·j ≤ f A·i
        using discretization-def by simp
      moreover from f-discr ‹(¬ B·(j+1) →t A·(i+n))›
      have f A·(i+n) ≤ f B·(j+1)
        using  discretization-def by simp
      ultimately show (f B·j ≤ f A·i ∧  f A·(i+n) ≤ f B·(j+1))
```

by *simp*
qed
ultimately have $\exists\ t\ i\ j\ f.\ quasiperiodic\ t$
$\wedge\ discretization\ f\ t$
$\wedge\ (f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1)\ )$
by *auto*
thus *False*
using $assms(5)$ **by** *simp*
qed

theorem *quasi-synchrony*:
assumes $card\ nodes = 2$
and $A \neq B$
and $T_{min} \geq 2 * \tau_{max}$
and $n > 0$
shows $\neg\ (\exists\ t\ i\ j\ f.$
$quasiperiodic\ t\ \wedge$
$discretization\ f\ t\ \wedge$
$(f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1)))$
$\longleftrightarrow n * T_{min} \geq T_{max} + 2{*}\tau_{max}$
proof
assume $\neg\ (\exists\ t\ i\ j\ f.$
$quasiperiodic\ t\ \wedge$
$discretization\ f\ t\ \wedge$
$(f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1)))$
with *assms* **show** $n * T_{min} \geq T_{max} + 2{*}\tau_{max}$
by (*rule quasi-synchrony-weakest*)
next
assume $n * T_{min} \geq T_{max} + 2{*}\tau_{max}$
show $\neg\ (\exists\ t\ i\ j\ f.$
$quasiperiodic\ t\ \wedge$
$discretization\ f\ t\ \wedge$
$(f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1)))$
proof
assume $(\exists\ t\ i\ j\ f.$
$quasiperiodic\ t\ \wedge$
$discretization\ f\ t\ \wedge$
$(f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i + n) \leq f\ B{\bullet}(j + 1)))$
then obtain $t\ i\ j\ f$
where $qp{:}quasiperiodic\ t$
and $discretization\ f\ t$
and $(f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i + n) \leq f\ B{\bullet}(j + 1))$
by *auto*
from $\langle discretization\ f\ t\rangle\ qp\ assms(2{-}4)\ \langle n * T_{min} \geq T_{max} + 2{*}\tau_{max}\rangle$
have $\neg\ (f\ B{\bullet}j \leq f\ A{\bullet}i \wedge f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1))$
by (*rule quasi-synchrony-sound*)
with $\langle(f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i + n) \leq f\ B{\bullet}(j + 1))\rangle$ **show** *False*
by *simp*
qed
qed

6 Quasi-Synchrony on Relaxed Communication

In this section we link the quasi-synchronous abstraction with quasi-periodic systems of, possibly, more than two nodes, assuming that there is no forbidden topologies in the communication graph.

6.1 Relaxed communication

We define here the notion of relaxed unitary discretization that only constrains events occuring on communicating nodes.

Now the happened-before relation only involves events that occur on communicating nodes.

definition *chb* (- $\hookrightarrow$- - [*100*, *100*] *100*)
  where $x \hookrightarrow t\ y$ == *com* (*node* x) (*node* y) $\wedge$ $x \rightarrow t\ y$

definition *relax-discr* ::
  (*event* $\Rightarrow$ *nat*) $\Rightarrow$ (*event* $\Rightarrow$ *tevent*) $\Rightarrow$ *bool*
  where *relax-discr* $f\ t$ = ($\forall$ $x\ y$. $f\ x < f\ y$ $\longleftrightarrow$ $x \hookrightarrow t\ y$)

lemma *f-relax-nhb*:
  assumes *relax-discr* $f\ t$
  and *com* A B
  and $f\ B{\bullet}j \leq f\ A{\bullet}i$
  shows $\neg$ $A{\bullet}i \hookrightarrow t\ B{\bullet}j$
  proof
    assume $A{\bullet}i \hookrightarrow t\ B{\bullet}j$
    with ‹*com* A B› ‹*relax-discr* $f\ t$› **have** $f\ A{\bullet}i < f\ B{\bullet}j$
      using *relax-discr-def chb-def* **by** *auto*
    thus *False*
      using ‹$f\ B{\bullet}j \leq f\ A{\bullet}i$› **by** *simp*
  qed

lemma *nhb-f-reflax*:
  assumes *relax-discr* $f\ t$
  and *com* A B
  and $\neg$ $A{\bullet}i \hookrightarrow t\ B{\bullet}j$
  shows $f\ B{\bullet}j \leq f\ A{\bullet}i$
  proof (*rule ccontr*)
    assume $\neg$ $f\ B{\bullet}j \leq f\ A{\bullet}i$
    hence $f\ B{\bullet}j > f\ A{\bullet}i$
      by *simp*
    with ‹*com* A B› *assms*(*1*) **have** $A{\bullet}i \hookrightarrow t\ B{\bullet}j$
      using *relax-discr-def chb-def* **by** *simp*
    thus *False*
      using ‹$\neg$ $A{\bullet}i \hookrightarrow t\ B{\bullet}j$› **by** *simp*
  qed

definition *tight-discr* ::
  (*event* $\Rightarrow$ *nat*) $\Rightarrow$ (*event* $\Rightarrow$ *tevent*) $\Rightarrow$ *bool*
  where *tight-discr* $f\ t$ =

$(relax\text{-}discr\ f\ t\ \wedge$
$(\forall\ A\ B\ i\ j.$
$com\ A\ B\ \wedge$
$f\ B{\bullet}j \leq f\ A{\bullet}i \wedge f\ A{\bullet}i < f\ B{\bullet}(j{+}1) \longrightarrow f\ A{\bullet}i = f\ B{\bullet}j))$

This is the most concise discretization. This property comes from the proof of the theorem on forbidden topologies (admitted here). If this discretization is not possible there exists an event $C{\bullet}k$ such that

$f\ B{\bullet}j < f\ C{\bullet}k \leq f\ A{\bullet}i$, that is, $B{\bullet}j \rightarrow 1\ C{\bullet}k \rightarrow 0\ A{\bullet}i$, or

$f\ B{\bullet}j \leq f\ C{\bullet}k < f\ A{\bullet}i$, that is, $B{\bullet}j \rightarrow 0\ C{\bullet}k \rightarrow 1\ A{\bullet}i$.

In both cases we get communication pattern C_0. In the following we assume that assumptions of the theorem on forbidden topologies holds. Hence for all quasi-periodic trace, there exists a tight discretization.

lemma *f-tight*:
assumes *tight-discr f t*
and *com A B*
and $A \neq B$
and $f\ B{\bullet}j \leq f\ A{\bullet}i$
and $f\ A{\bullet}i < f\ B{\bullet}(j{+}1)$
shows $f\ A{\bullet}i = f\ B{\bullet}j$
using *assms* **using** *tight-discr-def relax-discr-def* **by** *simp*

6.2 Soundness

lemma *not-qs-relax-sound-case1*:
assumes *qp*:*quasiperiodic t*
and $\forall\ C\ p\ q.\ p < q \longrightarrow arrival\ (t\ C{\bullet}p) < arrival\ (t\ C{\bullet}q)$
and $A \neq B$
and *com A B*
and $n > 0$
and $(\neg\ A{\bullet}i \hookrightarrow t\ B{\bullet}j)$
and $(A{\bullet}(i{+}n) \hookrightarrow t\ B{\bullet}(j{+}1))$
shows $n * T_{min} + \tau_{min} < T_{max} + \tau_{max}$
proof −
from *qp* **obtain** *transmin*: $\tau_{min} \leq trans\ (t\ A{\bullet}(i{+}n))$
and *transmax*: $trans\ (t\ A{\bullet}i) \leq \tau_{max}$
using *quasiperiodic-def* **by** *auto*
from *qp* $\langle A \neq B\rangle$ $\langle com\ A\ B\rangle$ $\langle\neg\ A{\bullet}i \hookrightarrow t\ B{\bullet}j\rangle$ **have** $arrival\ (t\ A{\bullet}i) > date\ (t\ B{\bullet}j)$
using *chb-def not-hb-realtime* **by** *simp*
hence $date\ (t\ B{\bullet}j) < date\ (t\ A{\bullet}i) + \tau_{max}$
using *transmax* **by** *simp*
moreover from *qp* $\langle A \neq B\rangle$ $\langle com\ A\ B\rangle$ *assms(2)* $\langle A{\bullet}(i{+}n) \hookrightarrow t\ B{\bullet}(j{+}1)\rangle$
have $date\ (t\ B{\bullet}(j{+}1)) \geq arrival\ (t\ A{\bullet}(i{+}n))$
using *chb-def hb-realtime* **by** *simp*
hence $date\ (t\ B{\bullet}(j{+}1)) \geq date\ (t\ A{\bullet}(i{+}n)) + \tau_{min}$
using *transmin* **by** *simp*
moreover from *qp* **have** $date\ (t\ B{\bullet}(j{+}1)) - date\ (t\ B{\bullet}j) \leq T_{max}$
using *Suc-eq-plus1 qp-suc* **by** *simp*
hence $date\ (t\ B{\bullet}(j{+}1)) \leq date\ (t\ B{\bullet}j) + T_{max}$
by *simp*

moreover have $n * T_{min} \leq$ *date* (*t A*•(*i*+*n*)) − *date* (*t A*•*i*)
using *qp* ‹*n* > *0*› **by** (*simp add*: *qp-cone-lower*)
ultimately show $n * T_{min} + \tau_{min} < T_{max} + \tau_{max}$
by *simp*
qed

lemma *not-qs-relax-sound-case2*:
assumes *qp*:*quasiperiodic t*
and ∀ *t C p q*. *p* < *q* ⟶ *arrival* (*t C*•*p*) < *arrival* (*t C*•*q*)
and *A* ≠ *B*
and *com A B*
and *n* > *0*
and (¬ *A*•*i* ↪*t B*•*j*)
and (*A*•(*i*+*n*) ↪*t B*•(*j*+*2*))
shows $n * T_{min} + \tau_{min} < 2 * T_{max} + \tau_{max}$
proof −
from *qp* **obtain** *transmin*: $\tau_{min} \leq$ *trans* (*t A*•(*i*+*n*))
and *transmax*: *trans* (*t A*•*i*) $\leq \tau_{max}$
using *quasiperiodic-def* **by** *auto*
from *qp* ‹*A* ≠ *B*› ‹*com A B*› ‹¬ *A*•*i* ↪*t B*•*j*› **have** *arrival* (*t A*•*i*) > *date* (*t B*•*j*)
using *chb-def not-hb-realtime* **by** *simp*
hence *date* (*t B*•*j*) < *date* (*t A*•*i*) + τ_{max}
using *transmax* **by** *simp*
moreover from *qp* ‹*A* ≠ *B*› ‹*com A B*› *assms*(*2*) ‹*A*•(*i*+*n*) ↪*t B*•(*j*+*2*)›
have *date* (*t B*•(*j*+*2*)) ≥ *arrival* (*t A*•(*i*+*n*))
using *chb-def hb-realtime* **by** *simp*
hence *date* (*t B*•(*j*+*2*)) ≥ *date* (*t A*•(*i*+*n*)) + τ_{min}
using *transmin* **by** *simp*
moreover from *qp* **have** *date* (*t B*•(*j*+*2*)) − *date* (*t B*•*j*) ≤ (*j*+*2* − *j*)$*T_{max}$
by (*metis Suc-eq-plus1 add.assoc add.commute add-diff-cancel-right′ qp-cone-upper*)
hence *date* (*t B*•(*j*+*2*)) − *date* (*t B*•*j*) ≤ $2*T_{max}$
by *simp*
hence *date* (*t B*•(*j*+*2*)) ≤ $2*T_{max}$ + *date* (*t B*•*j*)
by *simp*
moreover have $n * T_{min} \leq$ *date* (*t A*•(*i*+*n*)) − *date* (*t A*•*i*)
using *qp* ‹*n* > *0*› **by** (*simp add*: *qp-cone-lower*)
ultimately show $n * T_{min} + \tau_{min} < 2*T_{max} + \tau_{max}$
by *simp*
qed

lemma *qs-relax-sound*:
assumes *qp*:*quasiperiodic t*
and ∀ *t C p q*. *p* < *q* ⟶ *arrival* (*t C*•*p*) < *arrival* (*t C*•*q*)
and *relax-discr f t*
and *com A B*
and *A* ≠ *B*
and *n* > *0*
and $n * T_{min} + \tau_{min} \geq 2*T_{max} + \tau_{max}$
shows ¬ (*f B*•*j* ≤ *f A*•*i* ∧ *f A*•(*i*+*n*) ≤ *f B*•(*j*+*1*))
proof
assume (*f B*•*j* ≤ *f A*•*i* ∧ *f A*•(*i*+*n*) ≤ *f B*•(*j*+*1*))

hence $f\ B{\bullet}j \leq f\ A{\bullet}i$
and $f\ A{\bullet}(i+n) \leq f\ B{\bullet}(j+1)$
  by *simp-all*
moreover from $\langle n * T_{min} + \tau_{min} \geq 2*T_{max} + \tau_{max}\rangle$ *Tmaxpos'*
have $n * T_{min} + \tau_{min} \geq T_{max} + \tau_{max}$
  by *simp*
ultimately show *False*
proof (*cases* $f\ A{\bullet}(i+n) = f\ B{\bullet}(j+1)$)
  assume $f\ A{\bullet}(i+n) = f\ B{\bullet}(j+1)$
  have $B{\bullet}(j+1) \rightarrow t\ B{\bullet}(j+2)$
    by (*metis Suc-eq-plus1 add.commute add.left-commute hb1.simps lessI one-add-one tranclp.r-into-trancl*)
  with ⟨*relax-discr f t*⟩ **have** $f\ B{\bullet}(j+1) < f\ B{\bullet}(j+2)$
    using *relax-discr-def com-refl chb-def* **by** *simp*
  with $\langle f\ A{\bullet}(i+n) = f\ B{\bullet}(j+1)\rangle$ **have** $f\ A{\bullet}(i+n) < f\ B{\bullet}(j+2)$
    by *simp*
  with ⟨*relax-discr f t*⟩ ⟨*com A B*⟩ **have** $A{\bullet}(i+n) \hookrightarrow t\ B{\bullet}(j+2)$
    using *relax-discr-def chb-def* **by** *simp*
  moreover from ⟨*com A B*⟩ ⟨*relax-discr f t*⟩ $\langle f\ B{\bullet}j \leq f\ A{\bullet}i\rangle$ **have** $\neg\ A{\bullet}i \hookrightarrow t\ B{\bullet}j$
    by (*simp add*: *f-relax-nhb*)
  ultimately have $n * T_{min} + \tau_{min} < 2*T_{max} + \tau_{max}$
    using *assms* **by** (*simp add*: *not-qs-relax-sound-case2*)
  thus *False*
    using $\langle n * T_{min} + \tau_{min} \geq 2*T_{max} + \tau_{max}\rangle$ **by** *simp*
next
  assume $\neg\ f\ A{\bullet}(i+n) = f\ B{\bullet}(j+1)$
  with $\langle f\ A{\bullet}(i+n) \leq f\ B{\bullet}(j+1)\rangle$ **have** $f\ A{\bullet}(i+n) < f\ B{\bullet}(j+1)$
    by *simp*
  with ⟨*relax-discr f t*⟩ ⟨*com A B*⟩ **have** $A{\bullet}(i+n) \hookrightarrow t\ B{\bullet}(j+1)$
    using *relax-discr-def chb-def* **by** *simp*
  moreover from ⟨*com A B*⟩ ⟨*relax-discr f t*⟩ $\langle f\ B{\bullet}j \leq f\ A{\bullet}i\rangle$ **have** $\neg\ A{\bullet}i \hookrightarrow t\ B{\bullet}j$
    by (*simp add*: *f-relax-nhb*)
  ultimately have $n * T_{min} + \tau_{min} < T_{max} + \tau_{max}$
    using *assms* **by** (*simp add*: *not-qs-relax-sound-case1*)
  thus *False*
    using $\langle n * T_{min} + \tau_{min} \geq T_{max} + \tau_{max}\rangle$ **by** *simp*
qed
qed

6.3 Weakest condition

lemma *qs-relax-eg*:
assumes $\forall\ t\ C\ p\ q.\ p < q \longrightarrow arrival\ (t\ C{\bullet}p) < arrival\ (t\ C{\bullet}q)$
and $A \neq B$
and *com A B*
and $n > 0$
and $\neg\ (\exists\ t\ i\ j.\ quasiperiodic\ t$
  $\wedge\ (\neg\ A{\bullet}i \hookrightarrow t\ B{\bullet}j) \wedge (\neg\ A{\bullet}(i+n) \hookrightarrow t\ B{\bullet}(j+1)) \wedge (A{\bullet}(i+n) \hookrightarrow t\ B{\bullet}(j+2)))$
shows $n * T_{min} + \tau_{min} \geq 2*T_{max} + \tau_{max}$
proof (*rule ccontr*)
  assume $\neg\ n * T_{min} + \tau_{min} \geq 2*T_{max} + \tau_{max}$
  hence $\exists\ t\ i\ j.\ quasiperiodic\ t$
    $\wedge\ (\neg\ A{\bullet}i \hookrightarrow t\ B{\bullet}j) \wedge (\neg\ A{\bullet}(i+n) \hookrightarrow t\ B{\bullet}(j+1)) \wedge (A{\bullet}(i+n) \hookrightarrow t\ B{\bullet}(j+2))$

proof −
  from ‹¬ $n * T_{min} + \tau_{min} \geq 2*T_{max} + \tau_{max}$› **obtain** ε
    where $\varepsilon > 0$
    and $\varepsilon = 2*T_{max} + \tau_{max} - (n * T_{min} + \tau_{min})$
      by *simp*

  def *eg* == λe :: *event*.
            if node e = *A* ∧ *act e* = *0 then* ⦇ *date* = ε, *trans* = τ_{max} ⦈
        else if node e = *B then* ⦇ *date* = τ_{max} + *act e* ∗ T_{max}, *trans* = τ_{max} ⦈
        else ⦇ *date* = ε + *act e* ∗ T_{min}, *trans* = τ_{min} ⦈

  have ∀ *e*. *trans* (*eg e*) = τ_{min} ∨ *trans* (*eg e*) = τ_{max}
    using *eg-def* **by** *simp*
  hence ∀ *e*. τ_{min} ≤ *trans* (*eg e*) ∧ *trans* (*eg e*) ≤ τ_{max}
    using *eg-def taubounds* **by** *simp*
  moreover have ∀ *e*. *date* (*eg* (*step e*)) − *date* (*eg e*) = T_{min}
        ∨ *date* (*eg* (*step e*)) − *date* (*eg e*) = T_{max}
    using *eg-def* ‹$A \neq B$› **by** *simp*
  hence ∀ *e*. T_{min} ≤ *date* (*eg* (*step e*)) − *date* (*eg e*)
        ∧ *date* (*eg* (*step e*)) − *date* (*eg e*) ≤ T_{max}
    using *Tbounds* **by** *auto*
  moreover have ∀ *e*. *0* ≤ *date*(*eg e*)
    using *eg-def Tmaxpos Tminpos taumaxpos'* ‹$\varepsilon > 0$› **by** *simp*
  ultimately have *qp*:*quasiperiodic eg*
    using *quasiperiodic-def* **by** *simp*

  have *b0*:*date* (*eg B*•*0*) = τ_{max}
  and *b1*:*date* (*eg B*•*1*) = $T_{max} + \tau_{max}$
  and *b2*:*date* (*eg B*•*2*) = $2*T_{max} + \tau_{max}$
  and *a0*:*arrival* (*eg A*•*0*) = $\tau_{max} + \varepsilon$
  and *an*:*arrival* (*eg A*•*n*) = $\varepsilon + n * T_{min} + \tau_{min}$
    using *eg-def* ‹$A \neq B$› ‹$n > 0$› **by** *simp-all*

  from ‹$\varepsilon = 2*T_{max} + \tau_{max} - (n * T_{min} + \tau_{min})$›
  have *h*:$\varepsilon + n * T_{min} + \tau_{min} = 2*T_{max} + \tau_{max}$
    by *simp*

  from *h an b2* **have** *arrival* (*eg A*•*n*) ≤ *date* (*eg B*•*2*)
    by *simp*
  hence *A*•*n* ↪*eg B*•*2*
    using *chb-def* ‹*com A B*› *hb-arrival* **by** *auto*
  moreover from *h an b1 Tmaxpos* **have** *arrival* (*eg A*•*n*) > *date* (*eg B*•*1*)
    by *simp*
  with *qp assms*(*1*) ‹$A \neq B$› ‹*com A B*› **have** ¬ *A*•*n* ↪*eg B*•*1*
    using *chb-def* **by** (*metis hb-realtime not-le*)
  moreover from *a0 b0* ‹$\varepsilon > 0$› *taumaxpos* **have** *date* (*eg B*•*0*) < *arrival* (*eg A*•*0*)
    by *simp*
  with *qp assms*(*1*) ‹$A \neq B$› **have** ¬ *A*•*0* ↪*eg B*•*0*
    using *hb-realtime* **by** *force*
  ultimately have (¬ *A*•*0* ↪*eg B*•*0*) ∧ (¬ *A*•*n* ↪*eg B*•*1*) ∧ (*A*•*n* ↪*eg B*•*2*)
    by *simp*
  thus *?thesis*
    using *qp* **by** (*metis monoid-add-class.add.left-neutral*)

qed
with *assms*(*5*) **show** *False*
by *auto*
qed

lemma *qs-relax-weakest*:
assumes $\forall\ t.\ quasiperiodic\ t \longrightarrow (\exists\ f.\ tight\text{-}discr\ f\ t)$
and $\forall\ t\ C\ p\ q.\ p < q \longrightarrow arrival\ (t\ C{\bullet}p) < arrival\ (t\ C{\bullet}q)$
and $com\ A\ B$
and $A \neq B$
and $n > 0$
and $\neg\ (\exists\ t\ i\ j\ f.$
$quasiperiodic\ t\ \wedge$
$relax\text{-}discr\ f\ t\ \wedge$
$(f\ B{\bullet}j \leq f\ A{\bullet}i\ \wedge\ f\ A{\bullet}(i+n) \leq f\ B{\bullet}(j+1)))$
shows $n * T_{min} + \tau_{min} \geq 2{*}T_{max} + \tau_{max}$
proof (*rule ccontr*)
assume $\neg\ n * T_{min} + \tau_{min} \geq 2{*}T_{max} + \tau_{max}$
with *assms*(*2*−*5*)
have $\exists\ t\ i\ j.$
$quasiperiodic\ t\ \wedge$
$(\neg\ A{\bullet}i \hookrightarrow t\ B{\bullet}j) \wedge (\neg\ A{\bullet}(i+n) \hookrightarrow t\ B{\bullet}(j+1)) \wedge (A{\bullet}(i+n) \hookrightarrow t\ B{\bullet}(j+2))$
by (*metis qs-relax-eg*)
then obtain *t i j*
where *qp*:*quasiperiodic t*
and $(\neg\ A{\bullet}i \hookrightarrow t\ B{\bullet}j)$
and $(\neg\ A{\bullet}(i+n) \hookrightarrow t\ B{\bullet}(j+1))$
and $(A{\bullet}(i+n) \hookrightarrow t\ B{\bullet}(j+2))$
by *auto*
from *qp assms*(*1*) **obtain** *f*
where *tight-discr f t*
by *auto*
hence *f-discr*:*relax-discr f t*
using *tight-discr-def* **by** *simp*
have $f\ B{\bullet}j \leq f\ A{\bullet}i \wedge f\ A{\bullet}(i+n) \leq f\ B{\bullet}(j+1)$
proof −
from *f-discr* ‹$(A{\bullet}(i+n) \hookrightarrow t\ B{\bullet}(j+2))$› ‹$com\ A\ B$› **have** $f\ A{\bullet}(i+n) < f\ B{\bullet}(j+2)$
using *relax-discr-def chb-def* **by** *auto*
moreover from *f-discr* ‹$com\ A\ B$› ‹$\neg\ (A{\bullet}(i+n) \hookrightarrow t\ B{\bullet}(j+1))$›
have $f\ A{\bullet}(i+n) \geq f\ B{\bullet}(j+1)$
using *nhb-f-reflax* **by** *simp*
moreover have $B{\bullet}(j+1) \hookrightarrow t\ B{\bullet}(j+2)$
using *chb-def hb-subsequent com-refl* **by** (*simp add*:*tranclp.r-into-trancl*)
with *f-discr* **have** $f\ B{\bullet}(j+1) < f\ B{\bullet}(j+2)$
using *relax-discr-def chb-def com-refl* **by** *auto*
ultimately have $f\ A{\bullet}(i+n) = f\ B{\bullet}(j+1)$
using *f-tight* ‹$tight\text{-}discr\ f\ t$› ‹$A \neq B$› ‹$com\ A\ B$› **by** *simp*
moreover from *f-discr* ‹$com\ A\ B$› ‹$(\neg\ A{\bullet}i \hookrightarrow t\ B{\bullet}j)$› **have** $f\ B{\bullet}j \leq f\ A{\bullet}i$
by (*rule nhb-f-reflax*)
ultimately show $(f\ B{\bullet}j \leq f\ A{\bullet}i\ \wedge\ f\ A{\bullet}(i+n) \leq f\ B{\bullet}(j+1))$
by *simp*
qed

with *qp f-discr* **have** $\exists\ t\ i\ j\ f.\ quasiperiodic\ t$
$\wedge$ *relax-discr f t*
$\wedge\ (f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1)\ )$
by *auto*
thus *False*
using *assms*(*6*) **by** *auto*
qed

theorem *qs-relax*:
assumes $\forall\ t.\ quasiperiodic\ t \longrightarrow (\exists\ f.\ tight\text{-}discr\ f\ t)$
and $\forall\ t\ C\ p\ q.\ p < q \longrightarrow arrival\ (t\ C{\bullet}p) < arrival\ (t\ C{\bullet}q)$
and *com A B*
and $A \neq B$
and $n > 0$
shows $(\neg\ (\exists\ t\ i\ j\ f.\ quasiperiodic\ t\ \wedge$
relax-discr f t $\wedge$
$(f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1))))$
$\longleftrightarrow n * T_{min} + \tau_{min} \geq 2{*}T_{max} + \tau_{max}$
proof
assume $\neg\ (\exists\ t\ i\ j\ f.\ quasiperiodic\ t$
$\wedge$ *relax-discr f t*
$\wedge\ (f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1)))$
with *assms* **show** $n * T_{min} + \tau_{min} \geq 2{*}T_{max} + \tau_{max}$
by (*rule qs-relax-weakest*)
next
assume $n * T_{min} + \tau_{min} \geq 2{*}T_{max} + \tau_{max}$
show $\neg\ (\exists\ t\ i\ j\ f.\ quasiperiodic\ t$
$\wedge$ *relax-discr f t*
$\wedge\ (f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1)))$
proof
assume $(\exists\ t\ i\ j\ f.\ quasiperiodic\ t$
$\wedge$ *relax-discr f t*
$\wedge\ (f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1)))$
then obtain *t i j f*
where *qp*:*quasiperiodic t*
and *relax-discr f t*
and $(f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i + n) \leq f\ B{\bullet}(j + 1))$
by *auto*
with ‹*relax-discr f t*› *qp assms*(*2*−*5*) ‹$n * T_{min} + \tau_{min} \geq 2{*}T_{max} + \tau_{max}$›
have $\neg\ (f\ B{\bullet}j \leq f\ A{\bullet}i \wedge f\ A{\bullet}(i{+}n) \leq f\ B{\bullet}(j{+}1))$
using *qs-relax-sound* **by** *auto*
with ‹$(f\ B{\bullet}j \leq f\ A{\bullet}i \wedge\ f\ A{\bullet}(i + n) \leq f\ B{\bullet}(j + 1))$› **show** *False*
by *simp*
qed
qed

end