DTU Library

Technical Notes on the Proof of a Stutter Trace Inclusion Theorem

Haxthausen, Anne Elisabeth; Fantechi, Alessandro

Publication date:
2021

Document Version
Publisher's PDF, also known as Version of record

Technical Notes on the Proof of a Stutter Trace Inclusion Theorem

July 16, 2021

Anne E. Haxthausen[1] and Alessandro Fantechi[2]
[1]DTU Compute, Technical University of Denmark, Lyngby, Denmark
[2]DINFO, University of Florence, Via S. Marta 3, Firenze, Italy

Abstract. This document provides internal technical notes on our proof of a theorem about stutter trace inclusion stated in our manuscript entitled *Compositional verification of railway interlocking systems.* These notes are not standalone and rely on definitions given in that manuscript.

1 Introduction

Thoughout these notes, assume given a network N and two subnetworks N_1 and N_2 that have been created by a single cut through N according to our compositional method. Let m, m_1 and m_2 be models generated for these networks using the RobustRails tools for interlocking systems with the option without overlaps and without flank and front protection. Let $\overline{m|_i}$ be the reduced projection of m on network N_i and let $\overline{m_i}$ be the reduced model of m_i, for $i = 1, 2$.

In these notes we explain how we have proved the following theorem that is used for proving that our compositional verification method is sound.

1.1 The theorem

Theorem 1 (Reduced subnetwork model stutter trace includes reduced projection). *$\overline{m|_i} \trianglelefteq \overline{m_i}$, for $i = 1, 2$.*
(which means $\forall\ \overline{\pi|_i} \in Paths(\overline{m|_i})\ \ \exists\ \overline{\pi_i} \in Paths(\overline{m_i})$ such that $\overline{\pi|_i}$ and $\overline{\pi_i}$ are stutter equivalent).

1.2 Proof overview

We can prove Theorem 1 by proving that for an arbitrary path π in $Paths(m)$, it is possible to find a path π_i in $Paths(m_i)$, such that $\pi|_i$[1] and π_i are stutter equivalent wrt. the labeling functions of $\overline{m|_i}$ and $\overline{m_i}$, respectively. This approach is valid as by definition of the model projection and reduction operators, we have $Paths(\overline{m|_i}) = Paths(m|_i) = \{\pi|_i \mid \pi \in Paths(m)\}$ and $Paths(\overline{m_i}) = \{\pi_i \mid \pi_i \in Paths(m_i)\}$.

[1] Here the projection operator on states has been lifted to paths in the obvious way.

In Sec. 2 we will describe how to construct π_i from an arbitrary path π in $Paths(m)$ step by step by applying transition rules for m_i. At the same time we argue why the the applied transition rules are enabled. In order to argue for that some lemmas concerning properties of projected network elements and some theorems about relations between states in π and π_i are needed. These lemmas and theorems are defined in Sec. 3 and Sec. 4.

In Sec. 5 and Sec. 6 the path construction and state correspondence theorems are generalised to the case where several through routes are allowed through a cut.

In Sec. 7 we prove that $\pi|_i$ and π_i are stutter equivalent wrt. the labeling functions of $\overline{m|_i}$ and $\overline{m_i}$, respectively.

2 Construction of π_i

Given an arbitrary path π in $Paths(m)$, π_i is constructed as follows:

- The first state of π_i is chosen to be the initial state q_{i_0} of m_i.
- Then π_i is obtained by adding more and more states to its path by considering the transitions in π, one by one, in the order they appear. Each transition from a state q to a state q' in π, leads to the addition of 0, 1 or more states to π_i. What exactly should be added depends on the transition rule r that caused the state change from q to q' in π. Below we will explain this by case over various classes of rules.

We have the following classes of transition rules:

1. Some transition rules in m have **no counterpart in** m_i as they concern network elements in N for which the projection to N_i is undefined. A transition from a state q to a state q' in π caused by such a rule should not lead to any state change in π_i (so no state is added to π_i in our construction process). These rules include:
   (a) Rules for switching points p (i.e. changing $p.POS$) that are outside N_i.
   (b) Rules for switching signals s (i.e. changing $s.ACT$) that are outside N_i, **except if** s **is an entry signal for a through route** and thereby is projected to the added entry signal in N_i. For the exceptional case, see item 3a. Note that there is no exception, if s is an exit signal for a through route, even that it is mapped to the added exit signal in N_i, as that added exit signal is a border exit signal for which there are no transition rules.
   (c) Rules for train movements that only involve sections t outside N_i.
   (d) Rules for controllers of routes that are completely outside N_i.
2. Some transition rules in m **also exist in** m_i (with the same guards and same variable updates) as they only concern network elements in N for which the projection to N_i is the identity. A transition from a state q to a state q' in π caused by such a rule should lead to a state change in π_i obtained by applying the same rule to the last added state q_i. As the guard of such rule

was true for state q in π, it is also true for state q_i in π_i due to the fact that the guard only refers to variables for which there is a state correspondence according to Theorems 2, 3 and 4 (shown further below). These rules include:

(a) Rules for switching points p (i.e. changing $p.POS$) that are inside N_i (i.e. $p \in points(N_i)$)

(b) Rules for switching signals s (i.e. changing $s.ACT$) that are inside N_{-i} (i.e. $s \in signals(N_{-i})^2$)

(c) Rules describing the train movement of the head or tail of a train from a section t to a neighbouring section t', both inside N_i ($t, t' \in sections(N_i)$).

(d) Rules describing how the head or tail of a train enters/leaves the network at a border which is also in N_i.

(e) Rules for changing the mode of routes r completely inside N_i, except the *allocate* rule for the case where the route r is in conflict with one or several through routes cr. (This is an exeption, as the guard of the *allocate* rule for r will in that case have conditions on the states of conflicting through routes cr, but $proj_i$ is not the identity for such routes cr. Instead there is a corresponding rule, see item 3d.)

3. Some rules in m have a **corresponding rule for** m_i. A transition from a state q to a state q' in π caused by such a rule usually leads to a state change in π_i obtained by applying the corresponding rule to the last added state in π_i, however, in a few cases the application of the corresponding rule is deferred to a later step (where that rule is applied right after another rule in π_i).

   The rules having a corresponding rule include:

   (a) Rules for switching an entry signal s (changing $s.ACT$) of a through route for which s is outside N_i and therefore is mapped by the projection to an added entry signal s_{entry_i} in N_i: The corresponding rule is the one switching the added entry signal s_{entry_i}. If $s.ACT$ is changed to OPEN in π then the corresponding rule for opening the added entry signal s_{entry_i} should be applied in π_i (below we will explain that this is possible as the truth of its guard follows from the truth of the guard of the rule applied in π), but if the signal s is changed to CLOSED in π, the corresponding rule for closing the added entry signal s_{entry_i} should not be applied now (but later, see item 3i) in π_i. So when s is OPEN in the last considered state q of π, the added entry signal in N_i is also OPEN in last added state q_i of π_i.

   – Guards comparison: In the first case, we had that $CLOSED = s.ACT \neq s.CMD = OPEN$ in q (as the guard for opening the signal s was true in q). Let r be the route for which the $lock(r)$ rule had previously set $s.CMD = OPEN$ and $r.MODE = LOCKED$. $r.MODE$ must still be LOCKED as $s.ACT$ needs to be changed to $OPEN$ by the signal switching rule for s, before $r.MODE$ can be changed to $OCCUPIED$ by the $allocate(r)$ rule (otherwise a train can't enter the route and enable the $allocate(r)$ rule). The guard

[2] Note that $signals(N_{-i})$ does not include any added border signal present in N_i.

$s_{entry_i}.ACT \neq s_{entry_i}.CMD$ of the corresponding rule applied in π_i is also true for state q_i in π_i as $s_{entry_i}.CMD = s.CMD$ and $s_{entry_i}.ACT = s.ACT$ when $r.MODE = LOCKED$, cf. Theorem 6 formulas (3.0) and (4.0) for the case where there is only one through route and Theorem 10 formulas (3.0) and (4.0) for the case where there is several through routes.

(b) Any rule for rule for m describing the movement of the head or tail of **a train passing a cut** from t to t', where $t \in sections(N_i)$ and $t' \in sections(N) \setminus sections(N_i)$ is similar to the train movement rule for m_i describing the head or tail **leaving** the network N_i via t:
   i. The guards are the same, except for the movement of the head in the case where there is already a signal s in N protecting the entrance of t'. In this case there is an extra guard condition in m requiring this signal to be OPEN. (There is no such condition on the projected signal $proj_i(s)$ in m_i as this is an exit signal in N_i and its state is hence ignored by the exit rule for m_i.) When the guard of such rule is true for state q in π, the corresponding guard is also true for state q_i in π_i due to the fact that the corresponding guard is weaker and the common guard condition only refers to variable t for which there is a state correspondence according to Theorem 2.
   ii. Their updates are the same, except that in m_i there is no update of the occupancy status of t'. This also holds for $m|_i$ as t' is removed by the projection $proj_i$.

(c) Any rule for m describing the movement of the head or tail of **a train passing a cut** from t' to t, where $t \in sections(N_i)$ and $t' \in sections(N) \setminus sections(N_i)$ is similar to the train movement rule for m_i describing the head or tail **entering** the network N_i via t:
   i. Comparison of guards: (1) Such a rule for m has a guard condition on the occupancy status of the section t' that the train is leaving. That is not present in the guard of the corresponding rule for m_i and t' is removed by the projection. (2a) For head movements, if there is a signal s in N protecting the entrance of t, then there is an additional a guard condition in both rules requiring s to be OPEN. (2b) For head movements, if there is no signal in N protecting the entrance of t (so the train is using a through route r), then there is an extra guard condition in the corresponding rule for m_i requiring the added entry signal in N_i to be OPEN. Note that this added entry signal was opened when the entry signal of the through route was opened, cf. item 3a, and will first be closed in m_i when the $element\_in\_use(r, first(proj_i(r)))$ rule is applied in m, cf. item 3i, and that happens after the current step we are considering. So this extra guard condition in the corresponding rule requiring the added entry signal in N_i to be OPEN will be true. (3) For tail movements, in the corresponding rule for m_i there is extra guard condition on the occupancy status of t. The truth of that follows from the truth of the extra guard condition on the occupancy status of t' of the

rule for m, and train integrity (expressing reachable combinations of occupancy status of neighbouring sections).
Hence, when the guard of any rule for passing the cut in N is true for state q in π, the corresponding guard is also true for state q_i in π_i.

ii. Their updates are the same, except that in m_i there is no update of the occupancy status of t'. This also holds for $m|_i$ as t' is removed by the projection $proj_i$.

(d) The *allocate* rule for any route r which is totally inside N_i and which is in conflict with one or several through routes cr is similar to the *allocate* rule for r in m_i. When the former is applied in π, the latter should be applied in π_i. The guards of these rules are the same (r.MODE = MARKED and some conditions on the track sections of the route), except that conditions $(cr.MODE \neq ALLOCATING) \wedge (cr.MODE \neq LOCKED)$ for conflicting through routes cr of r in the former rule are replaced in the latter rule with conditions $(cr'.MODE \neq ALLOCATING) \wedge (cr'.MODE \neq LOCKED)$, where $cr' = proj_i(cr)$ is the projection of cr in N_i, as the set of conflicting rules of $proj(r) = r$ is exactly the set of projections of the conflicting routes cr of r, cf. Lemma 4. When the guard of the former rule is true in q, the guard of the latter rule is true in q_i due to the state correspondence stated in Theorems 2, 3 and 5. The variable updates of the two rules are exactly the same.

In the following, it is assumed that two through routes are not projected to the same route. This assumption will be removed in Sec. 5. Without loss of generality, assume that the through route r has direction UP, starts at a signal s_1 in N_{-1} and ends at a signal s_2 in N_{-2}, and let the first track section of the route path be t_1, as shown in Fig. 1.

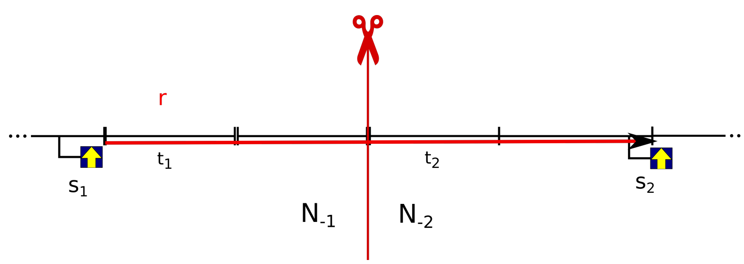

Fig. 1. A cut through a through route r in a network N.

(e) When the *dispatch* rule for a through route r is applied in π, the *dispatch* rule for $proj_i(r)$ should be applied in π_i. The guards of the two rules are the same modulo route renaming (they both require the route mode to be FREE). When the guard of the former rule is true in q, the guard

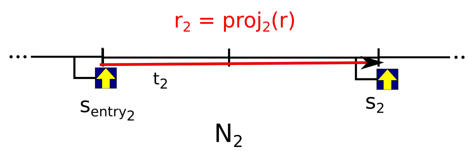

Fig. 2. Projected route r_2 of route r in Fig. 1.

of the latter rule is true in q_i due to the state correspondence stated in Theorem 5. Update of route modes are the same, modulo route renaming by $proj_i$.

(f) When the *allocate* rule for a through route r is applied in π, the *allocate* rule for $proj_i(r)$ should be applied in π_i. The conditions (on $e \in sections(r)$) in the guard of the former rule constitute a superset of the conditions in the guard of the latter rule and the conditions on the route modes are the same modulo renaming of routes (they both require the route mode to be MARKED). When the guard of the former rule is true in q, the guard of the latter rule is true in q_i due to the state correspondence stated in Theorem 2 (for sections and points) and Theorem 5 (for through routes). Update of route modes are the same, modulo renaming by $proj_i$. The updates of states of sections and points of the path in the rule for $proj_i(r)$ is a subset of the updates in the rule for r due to Lemmas 1 and 3. The additional updates in the rule for r are of sections and points that are removed by the projection.

(g) When the *lock* rule for a through route r is applied in π, the *lock* rule for $proj_i(r)$ should be applied in π_i. The conditions (on $e \in sections(r)$ and $p \in points(r)$) in the guard of the former rule constitute a superset of the conditions in the guard of the latter rule due to Lemmas 1 and 3, and the condition that the route mode is ALLOCATING is the same modulo route renaming. When the guard of the former rule is true in q, the guard of the latter rule is true in q_i due to the state correspondence stated in Theorems 2 (for sections and points), and 5 (for through routes). Updates of route mode and commanded setting (OPEN) of entry signals are the same, modulo route and signal renaming by $proj_i$.

(h) When the *route_in_use* rule for a through route r is applied in π, then, the *route_in_use* rule for $r_1 = proj_1(r)$ should be applied in π_1 (this rule is exactly the same modulo route and signal renaming by $proj_1$), while no state should be added to π_2. In the first case when the guard of the former rule is true in q, the guard of the latter rule is true in q_1 due to the state correspondence stated in Theorems 2 (for sections) and 5 (for through routes), and the updates of the route MODE (to OCCUPIED) and the route entry signals CMD variable $s_1.CMD$ (to CLOSED) are the same. $t_1.MODE$ is updated to the same (USED).

(i) When the *element_in_use*(r, e) rule for a through route r and a section e in the path of r (which is not the first section in the path) is applied in π, then, (1) if e is inside N_1 (and is not the first section t_1), then the $element\_in\_use(r_1, e)$ rule for $r_1 = proj_1(r)$ and e should be applied in π_1, but no state should be added to π_2; (2) else, if e is inside N_2 and is the first section t_2 in the path of $r_2 = proj_2(r)$, then first the $route\_in\_use(r_2)$ rule for $r_2 = proj_2(r)$ and e should be applied and then the signal rule closing the (added) entry signal s_{entry_2} of r_2 should be applied in π_2, and (3) otherwise (i.e. e is inside N_2 and is not the first section t_2 of r_2) then the $element\_in\_use(r_2, e)$ rule for $r_2 = proj_2(r)$ and e should be applied in π_2, but no state should be added to π_1.

Guard comparisons: The guard of the $element\_in\_use(r, e)$ has conditions on e (incl. $e.MODE = EXCLK$ and $\neg vacant(e)$) and its neighbour sections and requires $r.MODE = OCCUPIED$.
Case (1): The $element\_in\_use(r_1, e)$ rule has the same conditions on e and requires $r_1.MODE = OCCUPIED$. When the guard of the former rule is true in q, the guard of the latter rule is true in q_1 due to the state correspondence stated in Theorems 2 (for sections) and 5 formula (1.1) (for through routes) as $q(last(r_1)) \neq FREE$ as $e.MODE = EXCLK$ is true in q as the guard is true (which means that all sections after e, including $last(r_1)$ have MODE $EXCLK$).
Case (2): The guard of the $route\_in\_use(r_2)$ rule has conditions: $\neg vacant(e)$ and $r_2.MODE = LOCKED$. When the conditions of the guard of the rule applied in π are true in q, the former conditions are true in q_2 due to the state correspondence stated in Theorems 2 (for sections) and Theorem 5 formula (2.1) for through routes. After that rule has been applied the next rule can be applied because the commanded signal state now differs from the actual signal state.
Case (3): The guard of the $element\_in\_use(r_2, e)$ rule has the same conditions on e and requires $r_2.MODE = OCCUPIED$. When the guard of the $element\_in\_use(r, e)$ is true in q, the guard of $element\_in\_use(r_2, e)$ is true in q_2 due to the state correspondence stated in Theorems 2 (for sections) and Theorem 5 formula (2.2) for through routes as $q(first(r_2)) \neq EXCLK$ as the train has passed $first(r_2)$.

In all cases $e.MODE$ is updated to $USED$. In Case (2) the $route\_in\_use(r_2)$ rule additionally updates $r_2.MODE$ to $OCCUPIED$ and commands the source signal s_{entry_2} to be $CLOSED$. The second rule application then changes the actual signal $s_{entry_2}.ACT$ to $CLOSED$.

(j) When the $sequential\_release\_e$ rule for a through route r and section e (which is not the last section in the path of r) is applied in π, then (1) if e is not in N_i then no rule should be applied in π_i, (2) else if e is in N_i and is not the last element in the projected route $r_i = proj_i(r)$, then the $sequential\_release\_e$ rule for r_i and section e should be applied in π_i, and (3) otherwise (e is in N_1 and is the last element in the projected route $r_1 = proj_1(r)$), then the $release$ rule for r_1 should be applied in π_1. In cases (2) and (3), the guards of the two rules are identical modulo naming of routes, and when the guard of the former is true in q, it is also true in q_i due to the state correspondence rules. Both rules update $e.MODE$ to FREE. In case (3) the second rule additionally updates $r_1.MODE$ to FREE in π_1, while $r.MODE$ stays OCCUPIED in π.

(k) When the *release* rule for a through route r is applied in π, then, no rule should be applied in π_1 (as the last section of r is not in r_1 - the *release* rule for r_1 was already applied at the time where the *sequential_release_e* rule was applied in π for r and the last section in r_1, cf. the item above), while the *release* rule for $r_2 = proj_2(r)$ should be applied in π_2. In the latter case, the guards of the two rules are identical modulo naming of routes, and when the guard of the former is true in q, it is also true in q_2 due to the state correspondence rules.

The constructed state sequence π_i is a path of m_i as any path in $Paths(m_i)$ should start with q_{i_0} which is the case for π_i, and each time we added a new state q_i' to π_i as explained above we obtained that by applying a state transition rule of m_i to the latest added state q_i in π_i, and therefore $(q_i, q_i') \in R_i$, where R_i is the transition relation of m_i.

3 Lemmas about projection of network elements

In the following, let r be a route, $sections(r)$ be the set of sections in the path of r, $points(r)$ be the set of points in the path of r, $first(r)/last(r)$ be the first/last section in the path of r, $src(r)/dst(r)$ be the entry/exit signal of r, $req(r,p)$ be the required position of a point p in $points(r)$, and $conflicts(r)$ be the set of conflicting routes of r.

Lemma 1 (projection gives subsets of network elements). *$sections(proj_i(r)) \subseteq sections(r)$ and $points(proj_i(r)) \subseteq points(r)$when $proj_i(r)$ is defined.*

Lemma 2 (projection of entry/exit signals of a route). *$scr(proj_i(r)) = proj_i(src(r))$ and $dst(proj_i(r)) = proj_i(dst(r))$ when $proj_i(r)$ is defined.*

Lemma 3 (projection preserves required point settings). *Assume $proj_i(r)$ is defined. The required point setting for any point $p \in points(proj_i(r))$ in the path of $proj_i(r)$ in N_i is the same as for p in the path of r in N: $req(proj_i(r), p) = req(r, p)$.*

Lemma 4 (projection preserves and reflects conflicts).
$proj_i(cr) \in conflicts(proj_j(r))$ in N_i if and only if $cr \in conflicts(r)$ in N, when $proj_i(r), proj_i(cr)$ are defined and $proj_i(cr) \neq proj_i(r)$.

The lemmas follow from the definition of the projection function on network elements.

4 State correspondence theorems

Theorem 2 (State correspondence for track sections). *At any point in the construction process of π_i, it holds that $\overline{L_i}(q_i) = \overline{L|_i}(q|_i)$, where q is that last considered state in π and q_i is the last state added to π_i. Note that this also means that $q_i(t) = q(t)$ for sections $t \in sections(N_i)$.*

Proof by induction:
Base case: $\overline{L_i}(q_{i_0}) = \overline{L|_i}(q_0|_i)$, as the initial states of $\overline{m_i}$ and $\overline{m|_i}$ are the same, i.e. $q_{i_0} = q_0|_i$.
Induction step:
Assume that in the construction process of π_i we have considered state changes up to state q in π, and let the last state added to π_i be q_i. The induction hypothesis is that $\overline{L_i}(q_i) = \overline{L|_i}(q|_i)$. Now consider the next state transition in π from state q to a state q'. For that either no state is added to π_i or a state q_i' is added. In the first case it should be shown that $\overline{L_i}(q_i) = \overline{L|_i}(q'|_i)$, and in the second case it should be proved that $\overline{L_i}(q_i') = \overline{L|_i}(q'|_i)$.

For the three classes of transition rules we have.

1. The rule makes no changes to variables v for which $proj_i(v)$ is defined, so $q|_i = q'|_i$ and therefore $\overline{L|_i}(q|_i) = \overline{L|_i}(q'|_i)$. By the induction hypothesis we have $\overline{L_i}(q_i) = \overline{L|_i}(q|_i)$, so $\overline{L_i}(q_i) = \overline{L|_i}(q'|_i)$. Hence, this represents a stutter step and this case is proved.
2. The rule makes changes to variables v for which $proj_i(v)$ is defined, i.e. $q|_i \neq q'|_i$. The rule applied to π_i make the same changes to variables. As $\overline{L_i}(q_i) = \overline{L|_i}(q|_i)$ according to the induction hypothesis and the same changes are made to the same variables, it must also hold that $\overline{L_i}(q_i') = \overline{L|_i}(q'|_i)$.
3. (a) As $\overline{L_i}(q_i) = \overline{L|_i}(q|_i)$ according to the induction hypothesis, it must also hold that $\overline{L_i}(q_i') = \overline{L|_i}(q'|_i)$ as the two rules are only changing signals and these are removed by the reduction operation.
   (b) As $\overline{L_i}(q_i) = \overline{L|_i}(q|_i)$ according to the induction hypothesis and the same variable changes are made from q_i to q_i' as from $q|_i$ to $q'|_i$, (the variable t is changed in the same way and the variable t' that was changed from q to q' is removed by the projection $|_i$), it must also hold that $\overline{L_i}(q_i') = \overline{L|_i}(q'|_i)$.
   (c) As $\overline{L_i}(q_i) = \overline{L|_i}(q|_i)$ according to the induction hypothesis and the same variable changes are made from q_i to q_i' as from $q|_i$ to $q'|_i$, (the variable t is changed in the same way and the variale t' that was changed from q to q' is removed by the projection $|_i$), it must also hold that $\overline{L_i}(q_i') = \overline{L|_i}(q'|_i)$.
   (d) *allocate* for non through route: ok, as same changes of same section variables.
   (e) *dispatch* for through route: ok as there are no changes in section variables.
   (f) *allocate* for through route: ok as the section state updates are the same modulo sections removed by the projection.
   (g) *lock* for through route: ok as there are no changes in section variables
   (h) *route_in_use* for through route: For i = 1: Same argument as for item 2. For i = 2: : Same argument as for item 1.

(i) *element_in_use* for through route: Ok in all cases: either the changes of section variables are the same or the section variable changed in m is removed by the projection and there is no corresponding rule applied in m_i.
(j) *sequential_release_e* for through route: ok, same argument as above.
(k) *release* for through route: ok.

□

Theorem 3 (State correspondence for routes totally inside N_i). *At any point in the construction process of π_i, $q_i(r) = q(r)$ for those routes $r \in routes(N)$ that are completely inside N_i, where q is the last considered state in π and q_i is the last state added to π_i.*

Proof by induction: Consider a route $r \in routes(N)$ that is completely inside N_i.
Base case: The desired property holds for the intial states as in these all routes are in the same mode (FREE): $q_{i_0}(r) = q_0(r)$.
Induction step:
Assume that in the construction process of π_i we have considered state changes up to state q in π, and let the last state added to π_i be q_i. The induction hypothesis is that $q_i(r) = q(r)$. Now consider the next state transition in π from state q to a state q'. That will give rise to the addition of zero, one or more states in π_i. Let q'_i be the last added state by that. We want to prove that $q'_i(r) = q'(r)$. We only need to consider cases where the state transition in π is caused by a rule that changes the state (MODE) of r, (otherwise, the result is obvious). Such a rule belongs to one of the rule classes 2e and 3d. As $q_i(r) = q(r)$ according to the induction hypothesis and the same variable changes are made by these rules from q_i to q'_i as from q to q', it must also hold that $q'_i(r) = q'(r)$. □

Theorem 4 (State correspondence for signals that are not entry signals of through routes). *At any point in the construction process of π_i, $q_i(s) = q(s)$ for those signals $s \in signals(N)$ that are completely inside N_i and are not an entry signal of a through route, where q is the last considered state in π and q_i is the last state added to π_i.*

Proof by induction: Consider a signal s that is completely inside N_i and is not an entry signal of a through route.
Base case: The desired property holds for the intial states as in these all signals' commanded and actual settings are CLOSED: $q_{i_0}(s) = q_0(s)$.
Induction step:
Assume that in the construction process of π_i we have considered state changes up to state q in π, and let the last state added to π_i be q_i. The induction hypothesis is that $q_i(s) = q(s)$. Now consider the next state transition in π from state q to a state q'. We want to prove that $q'_i(s) = q'(s)$. We only need to consider cases where the state transition in π is caused by a rule that changes the state of $s.ACT$ or $s.CMD$, (otherwise, the result is obvious). The only rule that can change the state of $s.ACT$ in π belongs to class 2b. When that rule is

applied in π, the same rule is applied in π_i, so the changes to $s.ACT$ are the same in π and π_i. The only rules that can change $s.CMD$ in π are the *lock* and the *route_in_use* rules for routes r having s as entry signal (they will set $s.CMD$ to OPEN and CLOSED, respectively). For the two rules, r can't be a through route according to the assumption about s and is therefore totally inside N_i. Hence, these two rules will belong to class 2e and when these rules are applied in π, the same rules are applied in π_i, so the changes to $s.CMD$ are the same in π and π_i. □

Hence, the only variables v for which $q_i(proj_i(v))$ and $q(v)$ can differ, are variables for through routes and added entry signals. For these, we have the following state correspondence theorems.

Theorem 5 (State correspondence for through routes). *Let $r \in routes(N)$ be a through route and $r_i = proj_i(r)$ be its projection in N_i for $i = 1, 2$, where N_1/N_2 is the network that is on the same side of the cut as the first/last part of the route. It is assumed that r_1 and r_2 are not also equal to the projection of any other (through) route. (In later theorems, we drop these conditions.)*

At any point in the construction process of π_i, the following holds, where q is that last considered state in π and q_i is the last state added to π_i.

(1.0) $q_1(r_1.MODE) = q(r.MODE)$ when $q(r.MODE) \neq OCCUPIED$

(1.1) $q_1(r_1.MODE) = OCCUPIED$ when $q(r.MODE) = OCCUPIED$ and $q(last(r_1).MODE) \neq FREE$ (the condition expresses that r is occupied (partly of fully) by a train in N_1.)

(1.2) $q_1(r_1.MODE) = FREE$ when $q(r.MODE) = OCCUPIED$ and $q(last(r_1).MODE) = FREE$ (the condition expresses that r is occupied by a train which is not in N_1).

(2.0) $q_2(r_2.MODE) = q(r.MODE)$ when $q(r.MODE) \neq OCCUPIED$

(2.1) $q_2(r_2.MODE) = LOCKED$ when $q(r.MODE) = OCCUPIED$ and $q(first(r_2).MODE) = EXLCK$ (the condition expresses that r is occupied by a train which is not yet in N_2).

(2.2) $q_2(r_2.MODE) = OCCUPIED$ when $q(r.MODE) = OCCUPIED$ and $q(first(r_2).MODE) \neq EXLCK$ (the condition expresses that r is occupied by a train which is (partly or fully) in N_2).

Proof by induction:
Base case: The desired property holds for the intial states as in these all routes are in the same mode (FREE): $q_{i_0}(r_i) = q_0(r_i) = FREE$. Actually, it is state relations (1.0) and (2.0) that hold.

Induction step:
Assume that in the construction process of π_i we have considered state changes up to state q in π, and let the last state added to π_i be q_i. The induction hypothesis is that the stated property holds between q and q_i. Now consider the next state transition in π from state q to a state q', and let q'_i be the resulting last added state to π_i due to that step. We want to prove that the stated property holds between q' and q'_i. We only need to consider cases where the state transition in π is caused by a transition rule that changes the values of $r.MODE$, $last(r_1).MODE$ or $first(r_2).MODE$, or the associated state transition in π_i is caused by a transition rule that changes $r_i.MODE$, (otherwise, the result is obvious).

As the application of any of the $dispatch(r)$, $allocate(r)$, and $lock(r)$ rules in π leads to the application of $dispatch(r_i)$, $allocate(r_i)$, and $lock(r_i)$, respectively, in π_i, and these rules are applied in states for which $q_i(r_i.MODE) = q(r.MODE) \neq OCCUPIED$ (i.e. state relations (1.0) and (2.0) hold) and they make the same state changes of route modes in π and π_i, modulo route renaming by the projection functions, and the new route modes are still different from $OCCUPIED$, the state relations (1.0) and (2.0) are preserved for these applications.

Now we should consider cases where the application of the $route\_in\_use(r, e)$, $element\_in\_use(r, e)$, $sequential\_release\_e(r, e)$ and $release(r, e)$ rules in π together with associated changes in π_i make changes to the state relations.

π_1: When the $route\_in\_use(r, first(r))$ rule is applied in π, the $route\_in\_use(r_1, first(r))$ is applied in π_1 and both routes change mode to OCCUPIED, while $q(last(r_1).MODE)$ is still EXCLK - so now state relation (1.1) holds. In the step where the last section $last(r_1)$ of r_1 is released (i.e. $last(r_1).MODE$ becomes FREE) by $sequential\_release\_e(r, last(r_1))$ in π and by $release(r_1, last(r_1))$ in π_1, $r_1.MODE$ becomes FREE in π_1, but $r.MODE$ is still OCCUPIED in π (so the state relation becomes (1.2)). $r.MODE$ will first become FREE later when $release(r, last(r))$ is applied in π and nothing in π_1 (so the state relation becomes (1.0)).

π_2: When the $route\_in\_use(r, first(r))$ rule is applied in π (because a train had entered the first section $first(r)$ of r) $r.MODE$ will be changed from $LOCKED$ to $OCCUPIED$ and $first(r).MODE$ from EXLCK to USED in π, but no rule is applied in π_2, so $r_2.MODE$ stays $LOCKED$ - so now relation (2.1) holds. First when $element\_in\_use(r, first(r_2))$ is applied in π and $route\_in\_use(r_2, first(r_2))$ followed by the closing of the entry signal is applied in the same step in π_2 (because a train had entered the first section $first(r_2)$ of r_2 in both paths), $r_2.MODE$ will be changed from $LOCKED$ to $OCCUPIED$ in π_2 and $first(r_2).MODE$ will be changed to $USED$ both in π and π_2 - so now relation (2.2) holds. When $release(r)$ is applied in π and $release(r)$ in π_2, the state relation will change back to (2.0).

$\square$

The following theorem states that the state of signals in N and their projection in N_1 are the same thoughout the construction process, while this is not always the case for their projection in N_2, as the closing of the added entry signal is delayed.

Theorem 6 (State correspondence for entry signals of through routes). *Let s_1 be the entry signal of a through route r (as in Fig. 1), and let $s_1 = proj_1(s_1)$ and $s_{entry2} = proj_2(s)$ be its projections in N_1 and N_2, respectively, where N_1/N_2 is the network that is on the same side of the cut as the first/last part of the route. Also assume that s_{entry2} is not the projection of any other signal. (In a later theorem, we drop that assumption.) Let $r_2 = proj_2(r)$ and t_2 be the first section in r_2 (as in Fig. 1).*

At any point in the construction process of π_i, the following holds, where q is that last considered state in π and q_i is the last state added to π_i.

(1.0) $q_1(s_1.CMD) = q(s_1.CMD)$

(2.0) $q_1(s_1.ACT) = q(s_1.ACT)$

(3.0) $q_2(s_{entry2}.CMD) = q(s_1.CMD)$ *when* $q(r.MODE) \neq OCCUPIED$

(3.1) $q_2(s_{entry2}.CMD) = OPEN$
when $q(r.MODE) = OCCUPIED$ *and* $q(first(r_2).MODE) = EXLCK$
(the condition expresses that r is occupied by a train which is not yet in N_2)
(note that for this combination one can derive $q(s_1.CMD) = CLOSED$)

(3.2) $q_2(s_{entry2}.CMD) = q(s_1.CMD)(= CLOSED)$
when $q(r.MODE) = OCCUPIED$ *and* $q(first(r_2).MODE) \neq EXLCK$
(the condition expresses that r is occupied by a train which is (partly or fully) in N_2)

(4.0) $q_2(s_{entry2}.ACT) = q(s_1.ACT)$ *when* $q(r.MODE) \neq OCCUPIED$

(4.1a) $q_2(s_{entry2}.ACT) = q(s_1.ACT)(= OPEN)$ *when* $q(r.MODE) = OCCUPIED$ *and* $q(first(r_2).MODE) = EXLCK$, *and* $q(s_1.ACT) = OPEN$

(4.1b) $q_2(s_{entry2}.ACT) = OPEN$
when $q(r.MODE) = OCCUPIED$, $q(first(r_2).MODE) = EXLCK$, *and* $q(s_1.ACT) = CLOSED$
(the condition expresses that r is occupied by a train which is not yet in N_2 and the entry signal s_1 of r has been closed)

(4.2) $q_2(s_{entry2}.ACT) = q(s_1.ACT)(= CLOSED)$
when $q(r.MODE) = OCCUPIED \ \wedge \ q(first(r_2).MODE) \neq EXLCK$
(the condition expresses that r is occupied by a train which is (partly or fully) in N_2)

Proof by induction:

Base case: The desired property holds for the initial states. That follows from the following facts: In q_0: $s_1.CMD = s_1.ACT = CLOSED$ and $r.MODE = FREE \neq OCCUPIED$. In q_{1_0} : $s_1.CMD = s_1.ACT = CLOSED$. In q_{2_0} : $s_{entry_2}.CMD = s_{entry_2}.ACT = CLOSED$.

Induction step:

Assume that in the construction process of π_i we have considered state changes up to state q in π, and let the last state added to π_i be q_i. The induction hypothesis is that the stated relation holds between q and q_i. Now consider the next state transition in π from state q to a state q', and let q_i' be the resulting last added state to π_i due to that step. We want to prove that the stated relation holds between q' and q_i'.

The proof of the preservation of two first sub-relations (relating q_1 with q) is similar to the proof of the state correspondende for signals that are not entry signals of through routes. The only difference is that the *lock* and the *route_in_use* rules belong to classes 3g and 3h, respectively, and not to class 2e, but the conclusion for these rules is the same.

The proof for the remaining sub-relations (relating q_2 with q) is explained by considering the concurrent state transitions in π and π_2 that make changes to variables in subrelations (3.0-4.2). These will come in the order of the route life cycle for r:

1. Initially the conditions in (3.0) and (4.0) hold.
2. After concurrent dispatching/allocation of r and r_2, still the conditions in (3.0) and (4.0) hold.
3. When $lock(r)$ is applied in π and the concurrent $lock(r_2)$ is applied in π_2, $r.MODE$ is changed to $LOCKED(\neq OCCUPIED)$, $first(r_2).MODE$ to $EXCLK$ and $s_1.CMD$ to $OPEN$ in q, and $s_{entry2}.CMD$ is also changed to $OPEN$ in q_2. So still the conditions in (3.0) and (4.0) hold.
4. When $s_1.ACT$ is switched to OPEN in q, $s_{entry2}.ACT$ is also switched to OPEN in q_2. So still the conditions in (3.0) and (4.0) hold.
5. When the $route\_in\_use(r)$ is applied in π and nothing in π_2, $r.MODE$ is changed to OCCUPIED and $s_1.CMD$ to $CLOSED$ in q. So now the conditions for (3.1) and (4.1a) hold.
6. When $s_1.ACT$ is switched to CLOSED in q, $s_{entry2}.ACT$ is not changed. Now conditions in (3.1) and (4.1b) hold.
7. When $element\_in\_use(r, first(r_2))$ is applied in π and the concurrent $route\_in\_use(r_2)$ followed by switching rule for $s_{entry2}.ACT$ are applied in π_2, $first(r_2).MODE$ is changed to $USED(\neq EXCLK)$ in q and in q_2, and first $s_{entry2}.CMD$ and then $s_{entry2}.ACT$ are both changed to $CLOSED$ in q_2. Now conditions in (3.2) and (4.2) hold.
8. When the train has left $first(r_2)$ and the section is concurrently relased in π and π_2, $first(r_2).MODE$ is changed to $FREE$. The conditions in (3.2) and (4.2) still hold.
9. When $release(r)$ is applied in π and the concurrent $release(r_2)$ is applied in π_2, $r.MODE$ is changed to $FREE(\neq OCCUPIED)$ and the conditions in (3.0) and (4.0) hold again.

Consequences of the theorem are:
$q_2(s_{entry_2}.CMD) = OPEN$ when $q(s_1.CMD) = OPEN$
$q_2(s_{entry_2}.ACT) = OPEN$ when $q(s_1.ACT) = OPEN$.

In section 6, the state correpondence theorems for through routes and their entry signals will be generalised to cases where there are more than one through route.

5 Generalisation of the construction of π_i

We now generalise the rules for constructing π_i to cases where several ($n > 1$) through routes $r^1, \ldots, r^n$ are mapped to the same route by a projection.

Case 1 First we consider the case where several routes, $r^1, \ldots, r^n$, are mapped to the same route r_1 in N_1 by $proj_1$ ($r_1 = proj_1(r^j)$ for $j = 1, ..., n$) as shown in Fig. 3 for $n = 2$. .

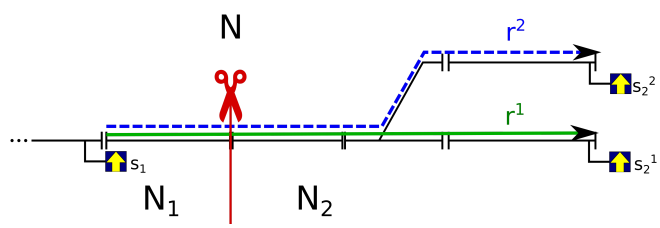

Fig. 3. Two up routes, r^1 and r^2, having the same projection r_1 in the down (left) subnetwork N_1.

For this case item 3e for dispatching of through routes must be adapted: When the $dispatch(r^j)$ rule for a through route r^j is applied in π, the $dispatch(r_1)$ rule for $r_1 = proj_1(r^j)$ should only be applied in π_1, if there is no other through route r^k that is in mode MARKED, ALLOCATED, LOCKED, or OCCUPIED with a train inside N_1 (i.e. r^k must be FREE or OCCUPIED with a train outside N_1), otherwise the application of the $dispatch(r_1)$ rule should be deferred: it is added to a queue of deferred route dispatchings.

Item 3j for sequential release of sections in through routes must be adapted for case (3):
When $sequential\_release\_e(r^j, e)$ is applied in π for some through route r^j and the last element e in the projected route $r_1 = proj_1(r)$, then first the $release(r_1)$ rule should be applied in π_1 and then, if there is any deferred route dispatching rule in the queue then that should be removed from the queue and applied as well in π_1 (which is possible as its guard requires the route mode to be FREE and the route mode was set to FREE by the previous step).

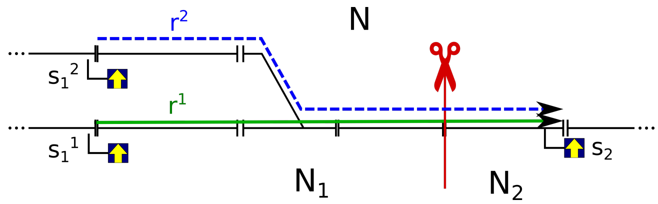

Fig. 4. Two up routes, r^1 and r^2, having the same projection r_2 in the up (right) subnetwork N_2.

Case 2 Then consider the case where several routes, $r^1, \ldots, r^n$, are mapped to the same route r_2 in N_2 by $proj_2$ ($r_2 = proj_2(r^j)$ for $j = 1, ..., n$) as shown in Fig. 4 for $n = 2$.

For this case item 3e for dispatching of through routes must be adapted: When the $dispatch(r^j)$ rule for a through route r^j is applied in π, the $dispatch(r_2)$ rule for $r_2 = proj_2(r^j)$ should only be applied in π_2, if all other through route r^k are in mode FREE, otherwise the application of the $dispatch(r_2)$ rule should be deferred: it is added to a queue of deferred route dispatchings.

Item 3k for sequential release of the last section in through routes must be adapted: When $release(r^k)$ is applied in π for some through route r^k, then first the $release(r_2)$ rule should be applied in π_2 and then, if there is any deferred route dispatching rule in the queue then that should be removed from the queue and applied as well in π_2 (which is possible as its guard requires the route mode to be FREE and the route mode was set to FREE by the previous step).

6 Generalised state correspondence

We now generalise the state correspondence theorems to cases where several ($n > 1$) through routes $r^1, \ldots, r^n$ are mapped to the same route by a projection.

6.1 Case 1

First we consider case 1 where several routes, $r^1, \ldots, r^n$, are mapped to the same route r_1 in N_1 by $proj_1$ as shown in Fig. 3 for $n = 2$: $r_1 = proj_1(r^j)$ for $j = 1, ..., n$. Let ${r^j}_2 = proj_2(r^j)$ and $s_{entry_2} = proj_2(s_1)$. Let $t_2 = first({r^j}_2)$ for $j = 1, ..., n$ (they all share the same first section.). Let $qs = \{q(r^j.MODE) | r^j \in \{r^1, \ldots, r^n\} \wedge q(r^j.MODE) \neq OCCUPIED\}$ be the set of states of those routes that are not OCCUPIED.

Theorem 7 (State correspondence for multiple through routes, case 1). *At any point in the construction process of π_i, the following holds, where q is that last considered state in π and q_i is the last state added to π_i.*

q_1 *is generalised:*

(1.0) $q_1(r_1.MODE) = max\{q(r^k.MODE)|r^k \in \{r^1, \ldots, r^n\}\}$, *when* $q(r^j.MODE) \neq OCCUPIED$ *for all* $r^j \in \{r^1, \ldots, r^n\}$ *(the condition expresses that no routes are OCCUPIED).*

(1.1) $q_1(r_1.MODE) = OCCUPIED$, *when there exists a route* $r^j \in \{r^1, \ldots, r^n\}$ *for which* $q(r^j.MODE) = OCCUPIED$ *and* $q(last(r_1).MODE) \neq FREE$ *(the condition expresses that one*[3] *of the routes* $r^1, \ldots, r^n$ *is occupied (partly of fully) by a train in* N_1*.)*

(1.2) $q_1(r_1.MODE) = FREE$, *when qs is empty and* $q(last(r_1).MODE) = FREE$ *(the condition expresses that all the routes* $r^1, \ldots, r^n$ *are OCCUPIED and no train is (anymore) in* N_1*).*

(1.3) $q_1(r_1.MODE) = max(qs)$, *when some routes, but not all routes are in mode OCCUPIED and* $q(last(r_1).MODE) = FREE$ *(the last condition expresses that no train is in* N_1*).*

q_2 *is defined as before, now for each of the through routes* $r^j \in \{r^1, \ldots, r^n\}$.

(2.0) $q_2(r^j{}_2.MODE) = q(r^j.MODE)$ *when* $q(r^j.MODE) \neq OCCUPIED$.

(2.1) $q_2(r^j{}_2.MODE) = LOCKED$ *when* $q(r^j.MODE) = OCCUPIED$ *and* $q(first(r^j{}_2).MODE) = EXLCK$.

(2.2) $q_2(r^j{}_2.MODE) = OCCUPIED$ *when* $q(r^j.MODE) = OCCUPIED$ *and* $q(first(r^j{}_2).MODE) \neq EXLCK$.

The new case (1.3) expresses that in the case where trains on all occupied routes have left N_1, and there are some remaining routes that are not in mode OCCUPIED, $r_1.MODE$ is the maximum mode of these unoccupied routes.

Proof: The proof is made by induction just as for Theorem 5, i.e. it is checked that the initial state satisfies the property and that the property is preserved by those transitions that change the variables used in the property.

Theorem 8 (State correspondence for the common entry signal of multiple through routes in case 1). *At any point in the construction process of* π_i*, the following holds, where* q *is that last considered state in* π *and* q_i *is the last state added to* π_i*.*

q_1 *is unchanged:*

(1.0) $q_1(s_1.CMD) = q(s_1.CMD)$

(2.0) $q_1(s_1.ACT) = q(s_1.ACT)$

[3] The formula just says that at least one route is occupied, but since there is a train in N_1 (on the common path of all the routes), only one route can be OCCUPIED).

q_2 is generalised:

(3.0) $q_2(s_{entry_2}.CMD) = q(s_1.CMD)$ when $\forall j \in \{1,\ldots,n\} : q(r^j.MODE) \neq OCCUPIED$ (i.e. no routes are occupied)

(3.1) $q_2(s_{entry_2}.CMD) = OPEN$ when $\exists j \in \{1,\ldots,n\} : q(r^j.MODE) = OCCUPIED$ and $q(t_2.MODE) = EXLCK$ (the condition expresses that some route r^k (note this need not to be the same as r^j) which is occupied by a train which has not yet entered N_2).

(3.2) $q_2(s_{entry_2}.CMD) = q(s_1.CMD)(= CLOSED)$ when $\exists j \in \{1,\ldots,n\} : q(r^j.MODE) = OCCUPIED$ and $q(t_2.MODE) \neq EXLCK$ (the condition expresses that at least one route is occupied by a train and all trains have (partly or fully) entered N_2).

(4.0) $q_2(s_{entry_2}.ACT) = q(s_1.ACT)$ when $\forall j \in \{1,\ldots,n\} : q(r^j.MODE) \neq OCCUPIED$

(4.1a) $q_2(s_{entry_2}.ACT) = q(s_1.ACT)(= OPEN)$ when $\exists j \in \{1,\ldots,n\} : q(r^j.MODE) = OCCUPIED$ and $q(t2.MODE) = EXLCK$, and $q(s_1.ACT) = OPEN$

(4.1b) $q_2(s_{entry_2}.ACT) = OPEN$ when $\exists j \in \{1,\ldots,n\} : q(r^j.MODE) = OCCUPIED$ and $q(t_2.MODE) = EXLCK$ and $q(s_1.ACT) = CLOSED$ (the conditions express that some route is occupied by a train which has not yet entered N_2 and the entry signal s_1 of that route has been closed)

(4.2) $q_2(s_{entry_2}.ACT) = q(s_1.ACT)(= CLOSED)$ when $\exists j \in \{1,\ldots,n\} : q(r^j.MODE) = OCCUPIED \wedge q(t_2.MODE) \neq EXLCK$ (the condition expresses that at least one route is occupied by a train and all trains have (partly or fully) entered N_2)

Proof: The proof is made by induction just as for Theorem 6, i.e. it is checked that the initial state satisfies the property and that the property is preserved by those transitions that change the variables used in the property.

6.2 Case 2

Then consider case 2 where several routes, $r^1,\ldots,r^n$, are mapped to the same route r_2 in N_2 by $proj_2$ as shown in Fig. 4 for $n = 2$: $r_2 = proj_2(r^j)$ for $j = 1,...,n$. Let ${r^j}_1 = proj_1(r^j)$. For this case we have $proj_1(s_1^j) = s_1^j$.

Note that at most one of the routes $r^1,\ldots,r^n$ can go through the states ALLOCATING, LOCKED and OCCUPIED at the same time.

Theorem 9 (State correspondence for multiple through routes, case 2). *At any point in the construction process of π_i, the following holds, where q is that last considered state in π and q_i is the last state added to π_i.*

q_1 is defined as before for each through route $r^j \in \{r^1, \ldots, r^n\}$:
(1.0) $q_1(r^j{}_1.MODE) = q(r^j.MODE)$ when $q(r^j.MODE) \neq OCCUPIED$

(1.1) $q_1(r^j{}_1.MODE) = OCCUPIED$ when $q(r^j.MODE) = OCCUPIED$ and $q(last(r^j{}_1).MODE) \neq FREE$ (the condition expresses that r^j is occupied (partly of fully) by a train in N_1.)

(1.2) $q_1(r^j{}_1.MODE) = FREE$ when $q(r^j.MODE) = OCCUPIED$ and $q(last(r^j{}_1).MODE) = FREE$ (the condition expresses that r^j is occupied by a train which is not in N_1).

q_2 is generalised:
(2.0) $q_2(r_2.MODE) = max(\{q(r^j.MODE)|r^j \in \{r^1, \ldots, r^n\}\})$ when $\forall j \in \{1, \ldots, n\}$: $q(r^j.MODE) \neq OCCUPIED$ (i.e. no routes are occupied)

(2.1) $q_2(r_2.MODE) = LOCKED$ when $\exists j \in \{1, \ldots, n\}$: $q(r^j.MODE) = OCCUPIED$ and $q(first(r_2).MODE) = EXLCK$
(the condition expresses that r^j is occupied by a train which has not yet entered N_2).

(2.2) $q_2(r_2.MODE) = OCCUPIED$ when $\exists j \in \{1, \ldots, n\}$: $q(r^j.MODE) = OCCUPIED$ and $q(first(r_2).MODE) \neq EXLCK$
(the condition expresses that r^j is occupied by a train which has (partly or fully) entered N_2).

Proof: The proof is made by induction just as for Theorem 5, i.e. it is checked that the initial state satisfies the property and that the property is preserved by those transitions that change the variables used in the property.

In the case where several signals are projected to the same added entry signal, the theorem for state correspondence of signals must be updated:

Theorem 10 (Generalised state correspondence for entry signals of through routes in case 2).
Let $s_1^1, \ldots, s_1^n$ be the entry signals in N_1 of the through routes $r^1, \ldots, r^n$ and $s_{entry_2} = proj_2(s_1^j)$ for $j = 1, \ldots, n$ be their common projection in N_2. By definition we have $proj_1(s_1^j) = s_1^j$ for $j = 1, \ldots, n$.

At any point in the construction process of π_i, the following holds, where q is that last considered state in π and q_i is the last state added to π_i.

q_1 is defined as before for each signal $s_1 \in \{s_1^1, \ldots, s_1^n\}$:
(1.0) $q_1(s_1.CMD) = q(s_1.CMD)$

(2.0) $q_1(s_1.ACT) = q(s_1.ACT)$

q_2 *is generalised:*

(3.0) $q_2(s_{entry_2}.CMD) = max(\{q(s_1^j.CMD) | s_1^j \in \{s_1^1, \ldots, s_1^n\}\})$ *when* $\forall j \in \{1, \ldots, n\} : q(r^j.MODE) \neq OCCUPIED$ *(i.e. no routes are occupied)*

(3.1) $q_2(s_{entry_2}.CMD) = OPEN$ *when* $\exists j \in \{1, \ldots, n\} : q(r^j.MODE) = OCCUPIED$ *and* $q(first(r_2).MODE) = EXLCK$ *(the condition expresses that* r^j *is occupied by a train which has not yet entered* N_2*).*

(3.2) $q_2(s_{entry_2}.CMD) = q(s_1^j.CMD)(= CLOSED)$ *when* $q(r^j.MODE) = OCCUPIED$ *and* $q(first(r_2).MODE) \neq EXLCK$ *(the condition expresses that* r^j *is occupied by a train which has (partly or fully) entered* N_2*).*

(4.0) $q_2(s_{entry_2}.ACT) = max(\{q(s_1^j.ACT) | s_1^j \in \{s_1^1, \ldots, s_1^n\}\})$ *when* $\forall j \in \{1, \ldots, n\} : q(r^j.MODE) \neq OCCUPIED$ *(i.e. no routes are occupied)*

(4.1a) $q_2(s_{entry_2}.ACT) = q(s_1^j.ACT)(= OPEN)$ *when* $q(r^j.MODE) = OCCUPIED$ *and* $q(first(r_2).MODE) = EXLCK$*, and* $q(s_1^j.ACT) = OPEN$

(4.1b) $q_2(s_{entry_2}.ACT) = OPEN$ *when* $\exists j \in \{1, \ldots, n\} : q(r^j.MODE) = OCCUPIED$ *and* $q(first(r_2).MODE) = EXLCK$ *and* $q(s_1^j.ACT) = CLOSED$ *(the conditions express that* r^j *is occupied by a train which has not yet entered* N_2 *and the entry signal* s_1^j *of that route has been closed)*

(4.2) $q_2(s_{entry_2}.ACT) = q(s_1^j.ACT)(= CLOSED)$
when $q(r^j.MODE) = OCCUPIED \ \wedge \ q(first(r_2).MODE) \neq EXLCK$
(the condition expresses that r^j *is occupied by a train which is (partly or fully) in* N_2*)*

Proof: The proof is made by induction just as for Theorem 6, i.e. it is checked that the initial state satisfies the property and that the property is preserved by those transitions that change the variables used in the property. The proof utilises the fact that at most one of the signals $s^1, \ldots, s^n$ can have their CMD/ACT variable to be OPEN at the same time, as at most one of the routes $r^1, \ldots, r^n$ can go through the states ALLOCATING, LOCKED and OCCUPIED at the same time.

Note that a consequence of this theorem is:
$q_2(s_{entry_2}.CMD) = OPEN$ when $\exists s \in \{{s_1}^1, ..., {s_1}^n\} : q(s.CMD) = OPEN$
$q_2(s_{entry_2}.ACT) = OPEN$ when $\exists s \in \{{s_1}^1, ..., {s_1}^n\} : q(s.ACT) = OPEN$

7 Proof of stutter equivalence

We have now proved that for an arbitrary path $\pi \in Path(m)$, we can construct paths $\pi_i \in Path(m_i)$ for $i = 1, 2$ such that they satisfy Theorem 2, i.e. $\overline{L_i}(q_i) =$

$\overline{L|_i}(q|_i)$, in any step of the construction process. Note that usually q has one corresponding state, but in a few cases (when deferred transitions are applied – for examples, see Sec. 6) two consecutive corresponding states. Similarly, several states q can have the same corresponding state.

Hence, the reduced (section) labels of states q in π are maintained by their corresponding states q_i in π_i and therefore the paths are stutter equivalent.