Agent Design with Goal Decomposition Trees: Companion Paper.

Bruno Mermet Gaële Simon Bruno Zanuttini *

July 15, 2009

1 Introduction and organization

This report is a companion for the article *Agent Design with Goal Decomposition Trees* [MSZ08]. We give here the detailed proofs that the verification step is valid, that is, that our proof schemas and propagation rules (contexts and GPFs) are correct.

Here we only recall the definitions and assumptions useful for proofs, in Section 2; Other definitions can be found in [MSZ08]. Section 2 also gives some useful lemmas. Then we proceed with each type of node successively. Leaf nodes are studied in Section 3, and the various operators in Sections 4–5. Finally, we study the case of lazy nodes in Section 6.

For each type of node, we recall the associated operational semantics, proof schemas, propagation rules for contexts (from the node to its children in the GDT), and propagation rules for GPFs (from the children to the node). We prove that if the obligations are verified, then executing the node terminates, and results in either the satisfaction condition or the GPF being true. We then prove that the propagation rule for contexts is correct. Since Section 6 contains every schema and proof related to lazy nodes, in all other sections we assume that the nodes are nonlazy.

2 Preliminaries

We first recall the following definitions from [MSZ08].

Definition 1 (NS/NNS, actions) *An action a is said to be* necessarily satisfiable *(NS) if for all worlds ω_a (resp. α_a) at which an agent ends (resp. starts) executing a, (α_a, ω'_a) satisfies $T'(post_a)$. For such an action a, we assume $gpf_a = \bot$.*

*GREYC, UMR CNRS 6072, Boulevard du Maréchal Juin, F-14032 Caen Cedex, France. E-mail addresses: bruno.mermet@univ-lehavre.fr, gsimon@iut.univ-lehavre.fr, bruno.zanuttini@info.unicaen.fr

Otherwise, a is said to be nonnecessarily satisfiable *(NNS). In this case, for all worlds α_a, ω_a as above, either a succeeds, and (α_a, ω'_a) satisfies $T'(post_a)$, or a fails and (α_a, ω'_a) satisfies $T'(gpf_a)$.*

Definition 2 (agent) *Let $\mathcal{E}$ be an environment. An* agent *A (in $\mathcal{E}$) is a tuple:*

$$(V_i(A), V_{\mathcal{E}}(A), \mathtt{init}_A, i_A, S_A, Actions_A, \mathtt{Beh}_A)$$

where $V_i(A)$ is a set of variables with $V_i(A) \cap V_{\mathcal{E}} = \emptyset$, $V_{\mathcal{E}}(A) \subseteq V_{\mathcal{E}}$, $\mathtt{init}_A$ is a mapping from $V_i(A)$ to values, $i_A \in \mathcal{L}_{V_i(A)}$, $S_A \subseteq \mathcal{L}_{V_i(A)}$ (and is finite), and $Actions_A$ is a set of actions whose preconditions, postconditions, and GPFs concern only variables in $V_{\mathcal{E}}(A) \cup V_i(A)$. and $\mathtt{Beh}_A$ is the behaviour of the agent. It is assumed that the interpretation of $V_i(A)$ as defined by $\mathtt{init}_A$ satisfies i_A.

Definition 3 (leaf node) *Let $\mathcal{E}$ be an environment, and let A be an agent. A* leaf node *N (of a GDT for A in $\mathcal{E}$) is a 6-tuple:*

$$(name_N, a_N, sc_N, gpf_N, lz_N, nsat_N)$$

where $a_N \in Actions_A$, $sc_N, gpf_N \in \mathcal{L}'_{V_{\mathcal{E}}(A) \cup V_i(A)}$, $lz_N \in \{L, NL\}$, and $nsat_N \in \{NS, NNS\}$. Moreover, if $lz_N = L$, then we must have $sc_N \in \mathcal{L}$ and if $nsat_N = NS$, then $gpf_N = \bot$.

Definition 4 (internal node) *Let A be an agent in an environment $\mathcal{E}$. An* internal node *N (of a GDT for A in $\mathcal{E}$) is a 7-tuple:*

$$(name_N, Op_N, Children_N, sc_N, gpf_N, lz_N, nsat_N)$$

where $sc_N, gpf_N, lz_N, nsat_N$ are as in Definition 3, Op_N is a decomposition operator, and $Children_N$ is a sequence of internal and leaf nodes whose length matches the arity of Op_N.

Definition 5 (LTL atoms) *Let N be a node in the GDT of an agent A, and let ω be an instant in the lifetime of A. Then $\omega \models init_N$ (resp. $\omega \models end_N$) if and only if A starts (resp. ends) executing N at instant ω, and $\omega \models in_N$ if and only if A starts, finishes, or is currently executing N.*

Now if $\omega \models in_N$, write α_N for the latest world before or equal to ω and satisfying $init_N$. Then $\omega \models sc_N$ if and only if the couple of interpretations (α_N, ω') satisfies sc_N; if $\omega \not\models in_N$, the value of sc_N at it is not defined. Finally, $\omega \models sat_N$ if and only if $\omega \models end_N \wedge sc_N$, and $\omega \models nonsat_N$ if and only if $\omega \models end_N \wedge \neg sc_N$.

Definition 6 (well-formed GDT) *A GDT is said to be* well-formed *if each of its nodes N satisfies the rules in Table 1, where $N_1, N_2, \ldots, N_n$ denote the children of N if it is an internal node.*

Definition 7 ($\alpha_N(\omega)$) *Let N be a node, and let ω be a world satisfying in_N. Then $\alpha_N(\omega)$ is defined to be the latest (i.e., maximal wrt $\prec$) world ω_b such that $\omega_b \preceq \omega$ and $\omega_b \models init_N$.*

All nodes	For $1 \leq i < j \leq n, \Box(\neg in_{N_i} \vee \neg in_{N_j})$	$(G.1)$
	For $1 \leq i \leq n, \Box(in_{N_i} \rightarrow in_N)$	$(G.2)$
	$\Box(\neg init_N \vee \neg end_N)$	$(G.3)$
	$\Box((in_N \wedge \neg end_N) \rightarrow \circ\neg init_N)$	$(G.4)$
Leaf nodes	$\Box(init_N \rightarrow \circ end_N)$	$(G.5)$
L nodes	$\Box(init_N \rightarrow \circ(sc_N \rightarrow end_N))$	$(G.6)$
	$\Box(init_N \rightarrow \circ(\neg sc_N \rightarrow init_N^{NL}))$	$(G.7)$

Table 1: Operational semantics for GDTs (Definition 6)

We now recall the following, fundamental assumption concerning the values of internal variables, and give two useful corollaries. In summary, the assumption states that internal variables of an agent change value only as the result of some action of this agent. The corollaries state that this cannot happen, in particular, right after the execution of some node terminates or right after the execution of some lazy node starts.

Assumption 8 (frame axiom) *Let A be an agent, and let $\varphi \in \mathcal{L}_{V_i(A)}$. If ω is a world such that ω satisfies φ and $\circ\omega$ does not satisfy φ, then this is the result of A beginning an action a at world ω (and finishing it at world $\circ\omega$) so that $\omega \models pre_a$ and either $\circ\omega \models post_a \models \neg\varphi$ or $\circ\omega \models gpf_a \models \neg\varphi$.*

Corollary 9 (frame axiom after the end of nodes) *Let N be a node in the GDT of Agent A. Then for all formulae φ such that $V(\varphi) \subseteq V_i(A)$ and for all worlds ω satisfying end_N, if ω satisfies φ, then so does $\circ\omega$.*

Proof Towards a contradiction, assume $\omega \models \varphi$ and $\circ\omega \not\models \varphi$. Then by Assumption 8 some action a must start at world ω. From the semantics of GDTs it follows that ω satisfies $init_N^{NL}$ for some leaf node N^{NL} with $a_N^{NL} = a$ and $lz_N^{NL} = NL$, or $\bullet\omega$ satisfies $init_N^L$ for some leaf node N^L with $a_N^L = a$, $lz_N^L = L$, and such that $\omega \not\models sc_N^L$ (Rule (G.7)).

In the first (nonlazy) case, we have that ω satisfies end_N and $init_N^{NL}$. From Rule (G.3) we get $N \neq N^{NL}$. Moreover, since N^{NL} is a leaf node, we get that it is a descendant of N in T. Finally, from Rules (G.3) and (G.5) we get $\omega \models end_N^{NL}$. Thus the execution of N terminates (strictly) before that of its descendant N^{NL}, in contradiction with the fact that signals *end.* go bottom up, and signals *init.* go top down in GDTs (Rule (G.2)).

In the lazy case, we have that ω satisfies end_N and $(init_N^L)^{NL}$ (Rule (G.7)), and we conclude as before. □

Corollary 10 (frame axiom for L nodes) *Let N be a lazy node in the GDT of Agent A. Let α_N be a world satisfying $init_N$ and $\circ\neg sc_N$. Then for all formulae φ such that $V(\varphi) \subseteq V_i(A)$, if α_N satisfies φ, then so does $\circ\alpha_N$.*

Proof The reasoning is similar to that in the proof of Corollary 9. Indeed, if α_N violates the claim, then some action a begins at this time, which can

only come as the result of a leaf node being executed at α_N (nonlazy leaf) or right before (lazy leaf), in both cases violating the fact that no signal is sent on initiating execution of a lazy node (Rules (G.6) and (G.7)). □

3 Leaf nodes

The operational semantics for leaf nodes only states that $init_N \rightarrow \circ end_N$ is always true (Rule (G.5)), which amounts to say that the duration of the associated action defines the duration of the leaf. In particular, since actions are supposed to terminate, we do not need to prove termination for leaf nodes.

Now the proof obligations are the following:

$$i_{\mathcal{E}} \wedge i_A \wedge c_N \models pre_a \quad (1)$$
$$i_{\mathcal{E}} \wedge i_A \wedge c_N \wedge T'(post_a) \models T'(sc_N) \quad (2)$$
$$i_{\mathcal{E}} \wedge i_A \wedge c_N \wedge T'(gpf_a) \models T'(gpf_N) \quad (3)$$

Proposition 11 (NL leaf nodes) *Let N be an NL leaf node, and let a be the associated action. Then if obligations 1 and 2 (resp. 1 and 3) are proven, execution of N succeeds (resp. fails) when a succeeds (resp. fails).*

Proof Let ω_N be a world satisfying end_N, and let $\alpha_N = \alpha_N(\omega_N)$. We have to prove that (α_N, ω'_N) satisfies $T'(sc_N)$ (resp. $T'(gpf_N)$) if a succeeds (resp. fails).

Obviously, α_N satisfies $i_{\mathcal{E}}$, i_A, and c_N. Thus Obligation 1 shows that it satisfies pre_a. It follows from Definition 1 that (α_N, ω'_N) satisfies $T'(post_a)$ when a succeeds (resp. $T'(gpf_a)$ when a fails), and the conclusion follows from Obligation 2 (resp. 3). □

As explained in [MSZ08], leaf nodes also come with the following obligations, which are enough to show that invariants and stable properties are correctly preserved in the whole GDT:

$$H_{elem} \models At'(i_{\mathcal{E}}) \quad (4)$$
$$H_{elem} \models At'(i_A) \quad (5)$$
$$H_{elem} \wedge s_{\mathcal{E}} \models At'(s_{\mathcal{E}}) \quad (6)$$
$$H_{elem} \wedge s_A \models At'(s_A) \quad (7)$$

with $H_{elem} = i_{\mathcal{E}} \wedge i_A \wedge c_N \wedge (T'(post_a) \vee T'(gpf_a))$.

Proposition 12 (environment) *Let N be a leaf node in the GDT of an agent A, and assume that the obligation generated from Schema (4) is proven. Then for all worlds α_N such that $\alpha_N \models init_N$, if α_N satisfies $i_{\mathcal{E}}$, then so does $\circ\alpha_N$. Similarly, for all stable properties $s_{\mathcal{E}} \in S_{\mathcal{E}}$, if the obligation generated from Schema (6) is proven, then for all worlds α_N such that $\alpha_N \models init_N$, if α_N satisfies $s_{\mathcal{E}}$, then so does $\circ\alpha_N$.*

$SeqAnd$	$\Box(sat_{N_1} \rightarrow \circ init_{N_2})$	$(SA.1)$
	$\Box(nonsat_{N_1} \rightarrow \circ end_N)$	$(SA.2)$
	$\Box(sat_{N_2} \rightarrow (end_N \wedge sat_N))$	$(SA.3)$
	$\Box(nonsat_{N_2} \rightarrow \circ end_N)$	$(SA.4)$
NL	$\Box(init_N \rightarrow init_{N_1})$	$(SA.5)$

Table 2: Operational semantics for *SeqAnd* (Definition 14)

Proof Obvious. □

Proposition 13 (agent) *Assume that for all elementary goals N in the GDT of an agent A the obligation generated from Schema (5) is proven. Then for all worlds ω in the trace of A, ω satisfies i_A. Similarly, for all stable properties $s_A \in S_A$, if the obligation generated from Schema (7) is proven for all nodes in the GDT of A, then for all worlds ω in its trace, if ω satisfies s_A, then so does $\circ\omega$.*

Proof Obvious from the fact that the invariant and stable properties of an agent only concern its internal variables (Definition 2) and Assumption 8. □

4 SeqAnd

Definition 14 (SeqAnd) *SeqAnd is the binary decomposition operator defined by the rules in Table 2, where N denotes the parent node with $Children_N = (N_1, N_2)$.*

The proof obligations for *SeqAnd* are the following:

$$i_{\mathcal{E}} \wedge \Sigma_{\mathcal{E}} \wedge i_A \wedge \Sigma_A \wedge c_N \wedge (T^{tmp}(sc_{N_1}))_{ri} \wedge T'_{tmp}(sc_{N_2}) \quad \models \quad T'(sc_N) \quad (8)$$

with $\Sigma_{\mathcal{E}} = \bigwedge_{s_{\mathcal{E}} \in S_{\mathcal{E}}}(s_{\mathcal{E}} \rightarrow At'(s_{\mathcal{E}}))$ and $\Sigma_A = \bigwedge_{s_A \in S_A}(s_A \rightarrow At'(s_A))$.

Proposition 15 (SeqAnd NL, termination) *Let N be a node with $Op_N = SeqAnd$, $Children_N = (N_1, N_2)$, and $lz_N = NL$. Then an execution of N terminates as soon as the corresponding executions of N_1 and N_2 do.*

Proof Let α_N be a world satisfying $init_N$. We have to show that α_N satisfies $\diamond end_N$.

From Rule (SA.5), we have $\alpha_N \models init_{N_1}$. Since N_1 terminates, there is a world ω_{N_1} such that $\omega_{N_1} \succeq \alpha_N$ and $\omega_{N_1} \models end_{N_1}$. Then if $\omega_{N_1} \models nonsat_{N_1}$, from Rule (SA.2) we get that it satisfies $\circ end_N$, which concludes. Otherwise we have $\omega_{N_1} \models sat_{N_1}$ (given the definition of atoms, see Definition 5), and from Rule (SA.1) we get that ω_{N_1} satisfies $\circ init_{N_2}$; as above we get a world $\omega_{N_2} \succeq \alpha_N$ and satisfying end_{N_2}. If it satisfies sat_{N_2}, then it satisfies end_N by Rule (SA.3), and otherwise it satisfies $\circ end_N$ by Rule (SA.4), which concludes in both cases. □

Proposition 16 (SeqAnd NL, correctness) *Let N be a node with $Op_N = SeqAnd$, $Children_N = (N_1, N_2)$, and $lz_N = NL$. Assume that proof obligation (8) is verified for N. Then an execution of N succeeds as soon as the corresponding executions of N_1 and N_2 terminate and succeed.*

Proof Let ω_N be a world satisfying end_N, and α_N be the latest world before ω_N and satisfying $init_N$. We have to show that (α_N, ω'_N) satisfies $T'(sc_N)$.

We have $\alpha_N \models init_{N_1}$ by Rule (SA.5). Let ω_{N_1} be the earliest world after α_N satisfying end_{N_1}, which exists since N_1 terminates. Since N_1 succeeds, we have that $(\alpha_N, \omega'_{N_1})$ satisfies $T'(sc_{N_1})$, and $\omega_{N_1} \models sat_{N_1}$. Now let α_{N_2} be $\circ\omega_{N_1}$. By Rule (SA.1), we have $\alpha_{N_2} \models init_{N_2}$. Like for N_1, there is an earliest world ω_{N_2} after α_{N_2} which satisfies end_{N_2} and sat_{N_2}. Thus $(\alpha_{N_2}, \omega'_{N_2})$ satisfies $T'(sc_{N_2})$.

Now it follows from the tree semantics of GDTs (Definition 6) that ω_{N_2} is exactly ω_N. Finally, summing up and translating world ω_{N_1} to the intermediate instant, we have:

$$(\alpha_N, \omega^{tmp}_{N_1}) \models T^{tmp}(sc_{N_1}) \quad (9)$$

$$(\alpha^{tmp}_{N_2}, \omega'_{N_2}) \models T'_{tmp}(sc_{N_2}) \quad (10)$$

Now from (9) we get the stronger $(\alpha_N, \omega^{tmp}_{N_1}) \models (T^{tmp}(sc_{N_1}))_{ri}$. From the frame axiom (Corollary 9) and from the fact that α_{N_2} is defined to be $\circ\omega_{N_1}$, we conclude $(\alpha_N, \alpha^{tmp}_{N_2}) \models (T^{tmp}(sc_{N_1}))_{ri}$. Finally, $(\alpha_N, \alpha^{tmp}_{N_2}, \omega'_{N_2})$ satisfies $(T^{tmp}(sc_{N_1}))_{ri} \wedge T'_{tmp}(sc_{N_2})$. Now by definition, α_N satisfies $i_{\mathcal{E}}$, i_A and c_N, $(\alpha_N, \omega'_{N_2})$ satisfies $\bigwedge_{s_{\mathcal{E}} \in S_{\mathcal{E}}}(s_{\mathcal{E}} \rightarrow At'(s_{\mathcal{E}}))$ and $\bigwedge_{s_A \in S_A}(s_A \rightarrow At'(s_A))$, and consequently, from the proof obligation, the above triple satisfies $T'(sc_N)$. Since this latter formula does not contain any variable of the form v^{tmp}, we finally have $(\alpha_N, \omega'_{N_2}) \models T'(sc_N)$, as desired since $\omega_N = \omega_{N_2}$. □

Proposition 17 (SeqAnd NL, GPF) *Let N be a node with $Op_N = SeqAnd$, $Children_N = (N_1, N_2)$, and $lz_N = NL$. Then:*

$$gpf_N \models (T'(gpf_{N_1}))_{ri} \vee ((T^{tmp}(sc_{N_1}))_{ri} \wedge (T'_{tmp}(gpf_{N_2}))_{ri})_{\ell r}$$

Proof Let ω_N be a world satisfying $nonsat_N$. From the circumscription assumption about LTL atoms and the definition of $nonsat_N$, we have that ω_N satisfies end_N and thus, that $\bullet\omega_N$ satisfies $nonsat_{N_1}$ (Rule (SA.2)) or $nonsat_{N_2}$ (Rule (SA.4)).

In the first case, by Definition $(\alpha_{N_1}, \bullet\omega'_N)$ satisfies $T'(gpf_{N_1})$. Moreover, from the tree semantics and hef act that only Rule (SA.5) can send Signal $init_{N_1}$, we have that α_{N_1} is exactly $\alpha_N(\omega_N)$. Thus $(\alpha_N(\omega_N), (\bullet\omega_N)')$ satisfies $T'(gpf_{N_1})$, and from Corollary 9 we conclude that $(\alpha_N(\omega_N), \omega'_N)$ satisfies $(T'(gpf_{N_1}))_{ri}$, as desired.

In the second case, it is easily shown that the corresponding execution of N_1 has succeeded and that of N_2 has failed. We thus have $(\alpha_{N_2}, (\bullet\omega_N)') \models T'(gpf_{N_2})$ and $(\alpha_{N_1}, \omega'_{N_1}) \models T'(sc_{N_1})$, and the second disjunct in the statement follows as above. □

Iter	$\Box(end_{N_1} \rightarrow \circ(\neg sc_N \rightarrow init_{N_1}))$ $\Box(end_{N_1} \rightarrow \circ(sc_N \rightarrow sat_N))$	$(I.1)$ $(I.2)$
NL	$\Box(init_N \rightarrow init_{N_1})$	$(I.3)$

Table 3: Operational semantics for *Iter* (Definition 20)

Proposition 18 (SeqAnd NL, context) *Let T be a GDT, and let N be a node in T with $Op_N = SeqAnd$, $Children_N = (N_1, N_2)$, and $lz_N = NL$. Then $c_{N_1} \models c_N$ and $c_{N_2} \models ((T'(sc_{N_1}))_{ri})_r$.*

Proof We first consider c_{N_1}. Let α_{N_1} be a world satisfying $init_{N_1}$. From the tree semantics of GDTs (Definition 6) we have that the only father of N_1 in T is N, and from the circumscription assumption about LTL atoms recalled in Section 2 it follows that only Rule (SA.5) can justify Signal $init_{N_1}$ being sent at α_{N_1}. Thus α_{N_1} satisfies $init_N$, and thus it satisfies c_N, as desired.

We now turn to c_{N_2}. The only rule able to send Signal $init_{N_2}$ is Rule (SA.1), from what it follows that $\bullet\alpha_{N_2}$ satisfies sat_{N_1}. Then from the Definitions of sat_{N_1} and sc_{N_1} (Definition 5) we get that $(\alpha_{N_1}, (\bullet\alpha_{N_2})')$ satisfies $T'(sc_{N_1})$, where α_{N_1} is the corresponding starting instant for N_1. Projecting onto internal variables on the right and using Corollary 9 we get that $(\alpha_{N_1}, \alpha'_{N_2})$ satisfies $(T'(sc_{N_1}))_{ri}$ and finally, projecting onto the right we get that α_{N_2} satisfies $((T'(sc_{N_1}))_{ri})_r$. □

5 Iter

Definition 19 (variant) *Let N be a node with $Op_N = Iter$ in the GDT of an agent A. Then a* variant *for N is a tuple $(v, <_v, v_0)$, where $v \in V_i(A)$ and $<_v$ is a total order on the values taken by v, such that every decreasing sequence of these values is lower-bounded by the value v_0.*

Definition 20 (Iter) *Iter is the unary decomposition operator defined by the rules in Table 3, where N denotes the parent node and N_1 denotes the child node.*

The proof obligations for *Iter* are the following:

$$\begin{aligned} H_{NL} &= i_{\mathcal{E}} \wedge \bigwedge_{s_{\mathcal{E}} \in S_{\mathcal{E}}} (s_{\mathcal{E}} \rightarrow At'(s_{\mathcal{E}})) \wedge i_A \wedge \bigwedge_{s_A \in S_A} (s_A \rightarrow At'(s_A)) \wedge c_N \\ H_1 &= (T'(sc_{N_1}))_{ri} \vee (T'(gpf_{N_1}))_{ri} \\ H_2 &= T^{tmp}(\neg sc_N) \wedge ((T'_{tmp}(sc_{N_1}))_{ri} \vee (T'_{tmp}(gpf_{N_1}))_{ri}) \end{aligned}$$

$$H_{NL} \wedge (H_1 \vee H_2) \wedge (v' = v_0) \models T'(sc_N) \quad (11)$$

$$H_{NL} \wedge H_1 \wedge (v' \neq v_0) \models v' <_v v \quad (12)$$

$$H_{NL} \wedge H_2 \wedge (v' \neq v_0) \models v' <_v v^{tmp} \quad (13)$$

Proposition 21 (Iter NL, termination) *Let N be a node with $Op_N = Iter$, $Children_N = (N_1)$, and $lz_N = NL$. Assume that proof obligations 11 to 13 are verified for N. Then an execution of N terminates as soon as the corresponding executions of N_1 do.*

Proof Let α_N be a world satisfying $init_N$. From Rule (I.3) we have $\alpha_N \models init_{N_1}$. Let $(\omega_{N_1})^1$ be the earliest world after α_N and satisfying end_{N_1}, which exists since N_1 terminates.

We distinguish two cases. First assume the value of v at $(\omega_{N_1})^1$ is v_0. Write ω_N for $\circ\omega^1_{N_1}$. Using Corollary 9 and the fact that v consists of internal variables only (Definition 19), we have $\omega_N \models (v = v_0)$. Moreover, depending on whether N_1 has succeeded or failed, $(\alpha_N, (\omega^1_{N_1})')$ satisfies $T'(sc_{N_1})$ or $T'(gpf_{N_1})$, and by Corollary 9 again, (α_N, ω'_N) satisfies $(T'(sc_{N_1}))_{ri}$ or $(T'(gpf_{N_1}))_{ri}$, that is, (α_N, ω'_N) satisfies hypothesis H_1. Finally, by definition of contexts and invariants, α_N satisfies c_N as well as $i_{\mathcal{E}}$ and i_A, and by definition of stable properties, (α_N, ω'_N) satisfies $\bigwedge_{s_{\mathcal{E}} \in S_{\mathcal{E}}}(s_{\mathcal{E}} \rightarrow At'(s_{\mathcal{E}}))$ and $\bigwedge_{s_A \in S_A}(s_A \rightarrow At'(s_A))$. It follows that (α_N, ω'_N) satisfies the hypotheses of Obligation 11 (with disjunct H_1), and thus it satisfies $T'(sc_N)$. Since $\omega^1_{N_1} \models end_{N_1}$ and $\omega_N = \circ\omega^1_{N_1}$, we get from Rule (I.2) that ω_N satisfies sat_N and thus end_N, as desired.

Now assume the value of v at $\omega^1_{N_1}$ is different from v_0. Then as above, from Obligation 12 it follows that it is less than its value at α_N. If $\omega^1_{N_1}$ satisfies $\circ sc_N$, then we conclude using Rule (I.2). Otherwise, let $\alpha^1_{N_1}$ be $\circ\omega^1_{N_1}$. By Rule (I.1) this world satisfies $init_{N_1}$. Moreover, by construction $(\alpha_N, (\alpha^1_{N_1})^{tmp})$ satisfies $T^{tmp}(\neg sc_N)$, and from Corollary 9 and the fact that v is over internal variables only, the value of v at $\alpha^1_{N_1}$ equals its value at $\omega^1_{N_1}$.

Thus, in case the execution of N does not end after the first iteration, by induction we get worlds $\omega^1_{N_1}, \alpha^1_{N_1}, \omega^2_{N_1}, \alpha^2_{N_1}, \ldots$ such that for all $i = 2, 3, \ldots$:

$$\begin{aligned} (\alpha_N, (\alpha^{i-1}_{N_1})^{tmp}) &\models T^{tmp}(\neg sc_N) \\ ((\alpha^{i-1}_{N_1})^{tmp}, (\omega^i_{N_1})') &\models (T'_{tmp}(sc_{N_1}))_{ri} \vee (T'_{tmp}(gpf_{N_1}))_{ri} \end{aligned}$$

Thus from Obligation 13 we get that the value of v at $\omega^i_{N_1}$ decreases with increasing i. It follows that for some i_0, $\omega^{i_0}_{N_1}$ satisfies $v = v_0$, and from the construction it thus satisfies:

$$\begin{aligned} (\alpha_N, (\alpha^{i_0-1}_{N_1})^{tmp}) &\models T^{tmp}(\neg sc_N) \\ ((\alpha^{i_0-1}_{N})^{tmp}, (\omega^{i_0}_{N_1})') &\models (T'_{tmp}(sc_{N_1}))_{ri} \vee (T'_{tmp}(gpf_{N_1}))_{ri} \\ (\omega^{i_0}_{N_1})' &\models v' = v_0 \end{aligned}$$

Now by Corollary 9 we get that $\circ\omega^{i_0}_{N_1}$ satisfies exactly the same hypotheses, that is, it satisfies the hypotheses of Obligation 11 (with disjunct H_2), and thus $(\alpha_N, (\circ\omega^i_{N_1})')$ satisfies $T'(sc_N)$. We conclude with Rule (I.2). □

Proposition 22 (Iter NL, correctness) *Let N be a node with $Op_N = Iter$, $Children_N = (N_1)$, and $lz_N = NL$. Assume that proof obligations 11 to 13 are verified for N. Then an execution of N succeeds as soon as the corresponding executions of N_1 terminate.*

Proof Obvious since by Proposition 21 the execution of N must end, but by Definition of *Iter* the signal $nonsat_N$ cannot be sent. □

Proposition 23 (Iter NL, GPF) *Let N be a node with $Op_N = Iter$. Then $gpf_N \models \bot$.*

Proof Direct from the fact that N always succeeds (Proposition 22) and the definition of GPFs. □

Proposition 24 (Iter NL, context) *Let T be a GDT, and let N be a node in T with $Op_N = Iter$, $Children_N = (N_1)$, and $lz_N = NL$. Then $c_{N_1} \models c_N \vee (\neg T'(sc_N))_r$.*

Proof From the circumscription assumption about LTL atoms and the rules defining *Iter* it follows that any world α_{N_1} satisfying $init_{N_1}$ either satisfies $init_N$ or is such that $\bullet\alpha_{N_1} \models end_{N_1}$ and $\alpha_{N_1} \models \neg sat_N$, hence the result. □

6 Lazy nodes

As explained in [MSZ08], the proofs schemas and propagation rules for a lazy node N can be derived from the ones for its nonlazy counterpart N^{NL}. Also recall that the operational semantics of lazy node is given by $\Box(init_N \rightarrow \circ(sc_N \rightarrow end_N))$ (Rule (G.6)) and $\Box(init_N \rightarrow \circ(\neg sc_N \rightarrow init_N^{NL}))$ (Rule (G.7)).

Proposition 25 (lazy nodes, proof schemas) *Let N be a lazy node, and let N^{NL} be the same node except for $lz_{N^{NL}} = NL$. Define c_N^{NL} to be $\neg sc_N \wedge (c_N)_i$. Then if all proof obligations generated for N^{NL} with this context, and the additional obligation $(T'(gpf_N))_{\ell i} \models T'(gpf_N)$, are proven, N terminates and succeeds (resp. establishes its GPF) when its satisfaction condition is initially true or the corresponding execution of N^{NL} terminates and succeeds (resp. establishes it GPF).*

Proof Let ω_N be a world satisfying end_N, and $\alpha_N = \alpha_{\omega_N}(N)$. First, if α_N satisfies $\circ sc_N$, then by Rule (G.6) we get $\omega_N = \circ\alpha_N$, and by the same rule together with the definition of LTL atom sc_N we have $\omega_N \models sc_N$. Hence the execution of N terminates (immediately) and succeeds.

Now assume that α_N does not satisfy $\circ sc_N$. Then by Rule (G.7) we get $\circ\alpha_N \models init_N^{NL}$. Moreover, by definition of contexts we have that $\alpha_N \models c_N$, thus *a fortiori* $\alpha_N \models (c_N)_i$ and by Corollary 10 we get $\circ\alpha_N \models (c_N)_i$. Write ω_N^{NL} for the first world after (or equal to) $\circ\alpha_N$ and satisfying end_N^{NL}. This world exists since the execution of N^{NL} is assumed to terminate. Then by the operational semantics of laziness we have $\omega_N^{NL} = \omega_N$.

First assume that the execution of N^{NL} succeeds. Then from the definition of proof obligations and the assumption we have that $(\circ\alpha_N, \omega'_N)$ satisfies $T'(sc_N^{NL})$. But since satisfaction conditions of lazy nodes are in $\mathcal{L}$ (Definition 4), this is equivalent to $\omega_N \models sc_N^{NL} = sc_N$, hence N succeeds.

Finally, assume that the execution of N^{NL} establishes its GPF. Then we have that $(\circ\alpha_N, \omega'_N)$ satisfies $T'(gpf_N^{NL}) = T'(gpf_N)$. From Corollary 10 it follows that (α_N, ω'_N) satisfies $(T'(gpf_N))_{\ell i}$ and thus, by assumption it satisfies $T'(gpf_N)$, as desired. □

Proposition 26 (lazy nodes, context) *Let N be a lazy node, and let N^{NL} be the same node except for $lz_{N^{NL}} = NL$. Define c_N^{NL} to be $\neg sc_N \wedge (c_N)_i$. Then for all nodes $N_i \in Children_N$, c_{N_i} entails $c_{N_i}^{NL}$, where $c_{N_i}^{NL}$ is the context of N_i as computed from N^{NL} using c_N^{NL}.*

Proof From the tree semantics of GDTs and Rule (G.7) we first get that if $\omega \models init_{N_i}$ for some world ω, then this corresponds to an execution of N starting at α_N with $\alpha_N \not\models \circ sc_N$. Using the same reasoning as in the proof of Proposition 25 we get that the decomposition of N gets executed as if N were nonlazy, but with context $\neg sc_N \wedge (c_N)_i$, hence the result. □

To conclude, as concerns preservation of invariant and stable properties by leaf nodes, the proof schemas are the same as those given in Section 3, but with hypotheses $H_{elem} = i_{\mathcal{E}} \wedge i_A \wedge \neg sc_N \wedge (c_N)_i \wedge (T'(post_a) \vee T'(gpf_a))$. The proof that these schemas are correct is straightforward.

References

[MSZ08] Bruno Mermet, Gaële Simon, and Bruno Zanuttini. Agent Design with Goal Decomposition Trees. Technical report, GREYC-UMR 6072, 2008.

$SeqOr$	$\Box(\neg in_{N_1} \vee \neg in_{N_2})$	$(SO.1)$
	$\Box((in_{N_1} \vee in_{N_2}) \rightarrow in_N)$	$(SO.2)$
	$\Box((end_{N_1} \wedge sat_{N_1}) \rightarrow (end_N \wedge sat_N))$	$(SO.3)$
	$\Box((end_{N_1} \wedge \neg sat_{N_1}) \rightarrow \circ init_{N_2})$	$(SO.4)$
	$\Box((end_{N_2} \wedge sat_{N_2}) \rightarrow (end_N \wedge sat_N))$	$(SO.5)$
	$\Box((end_{N_2} \wedge \neg sat_{N_2}) \rightarrow \circ end_N)$	$(SO.6)$
NL	$\Box(init_N \rightarrow init_{N_1})$	$(SO.7)$
L	$\Box(init_N \rightarrow \neg in_{N_1} \wedge \neg in_{N_2} \wedge \circ(\neg sat_N \rightarrow init_{N_1}))$	$(SO.8)$

$Case(.,.)$	$\Box(\neg in_{N_1} \vee \neg in_{N_2})$	$(C.1)$
	$\Box((in_{N_1} \vee in_{N_2}) \rightarrow in_N)$	$(C.2)$
	$\Box(cond_{N_1} \vee cond_{N_2})$	$(C.3)$
	$\Box((end_{N_1} \wedge sat_{N_1}) \rightarrow (end_N \wedge sat_N))$	$(C.4)$
	$\Box((end_{N_1} \wedge \neg sat_{N_1}) \rightarrow \circ end_N)$	$(C.5)$
	$\Box((end_{N_2} \wedge sat_{N_2}) \rightarrow (end_N \wedge sat_N))$	$(C.6)$
	$\Box((end_{N_2} \wedge \neg sat_{N_2}) \rightarrow \circ end_N)$	$(C.7)$
NL	$\Box(init_N \rightarrow \circ(init_{N_1} \vee init_{N_2}))$	$(C.8)$
	$\Box(init_N \rightarrow \circ(\neg cond_{N_1} \rightarrow init_{N_2}))$	$(C.9)$
	$\Box(init_N \rightarrow \circ(\neg cond_{N_2} \rightarrow init_{N_1}))$	$(C.10)$
L	$\Box(init_N \rightarrow \neg in_{N_1} \wedge \neg in_{N_2} \wedge \circ(\neg sat_N \rightarrow \circ(init_{N_1} \vee init_{N_2})))$	$(C.11)$
	$\Box(init_N \rightarrow \circ(\neg sat_N \rightarrow \circ(\neg cond_{N_1} \rightarrow init_{N_2})))$	$(C.12)$
	$\Box(init_N \rightarrow \circ(\neg sat_N \rightarrow \circ(\neg cond_{N_2} \rightarrow init_{N_1})))$	$(C.13)$

And	$\Box(\neg in_{N_1} \vee \neg in_{N_2})$	$(A.1)$
	$\Box((in_{N_1} \vee in_{N_2}) \rightarrow in_N)$	$(A.2)$
	$\Box(init_N \rightarrow \neg second_N)$	$(A.3)$
	$\Box((second_N \wedge in_N) \rightarrow \circ second_N)$	$(A.4)$
	$\Box((\neg second_N \wedge \neg end_{N_1} \wedge \neg end_{N_2}) \rightarrow \circ \neg second_N)$	$(A.5)$
	$\Box((end_{N_1} \wedge second_N \wedge sat_{N_1}) \rightarrow (end_N \wedge sat_N))$	$(A.6)$
	$\Box((end_{N_1} \wedge \neg second_N \wedge sat_{N_1}) \rightarrow \circ(init_{N_2} \wedge second_N))$	$(A.7)$
	$\Box((end_{N_1} \wedge \neg sat_{N_1}) \rightarrow \circ end_N)$	$(A.8)$
	$\Box((end_{N_2} \wedge second_N \wedge sat_{N_2}) \rightarrow (end_N \wedge sat_N))$	$(A.9)$
	$\Box((end_{N_2} \wedge \neg second_N \wedge sat_{N_2}) \rightarrow \circ(init_{N_1} \wedge second_N))$	$(A.10)$
	$\Box((end_{N_2} \wedge \neg sat_{N_2}) \rightarrow \circ end_N)$	$(A.11)$
NL	$\Box(init_N \rightarrow (init_{N_1} \vee init_{N_2}))$	$(A.12)$
L	$\Box(init_N \rightarrow \neg in_{N_1} \wedge \neg in_{N_2} \wedge \circ(\neg sat_N \rightarrow (init_{N_1} \vee init_{N_2})))$	$(A.13)$