

Bachelorarbeit. ACHTUNG NICHT FERTIG!

Thure Nebendahl

May 23, 2020

Contents

Adding Operators. ACHTUNG NICHT FERTIG!

In this chapter we will extend the algorithm presented in [?] by the operators $\Box\phi$ (always), $\Box^-\phi$ (always in the past), $\Diamond\phi$ (eventually), $\Diamond^-\phi$ (some time in the past) that were already presented in the paper but not considered when defining the algorithm. We are extending the algorithm because these operators are necessary for a lot of relevant queries. First we will show how to modify the definitions of $\text{eval}^n(\alpha)$, $\Phi_0(\psi)$ and $\Phi_i(\psi)$, before we will continue to prove why the modifications are correct. As a reminder the semantics of these four temporal queries (TQs) are defined as follows:

Definition 0.1 (semantics of TQs cf. Definition 3.3 in [?]). Let ϕ be a TQ, $\mathcal{I} = (I_i)_{0 \leq i \leq n}$ a sequence of interpretations over a common domain, $\mathbf{a} : \text{FVar}(\phi) \rightarrow \mathbf{N}_{\mathbb{C}}$ a variable assignment, and i be an integer with $0 \leq i \leq n$. The *satisfaction relation* $\mathcal{I}, i \models \mathbf{a}(\phi)$ is defined by induction on the structure of ϕ as follows:

ϕ	$\mathcal{I}, i \models \mathbf{a}(\phi)$
$\Box\phi_1$	$\mathcal{I}, k \models \mathbf{a}(\phi_1)$ for all $k, i \leq k \leq n$
$\Box^-\phi_1$	$\mathcal{I}, k \models \mathbf{a}(\phi_1)$ for all $k, 0 \leq k \leq i$
$\Diamond\phi_1$	$\mathcal{I}, k \models \mathbf{a}(\phi_1)$ for some $k, i \leq k \leq n$
$\Diamond^-\phi_1$	$\mathcal{I}, k \models \mathbf{a}(\phi_1)$ for some $k, 0 \leq k \leq i$

Table 1: semantics of TQs

$\text{FVar}(\phi)$ denotes the set of *free variables* of a TQ and is defined as the union of the sets $\text{FVar}(\psi)$ of all queries ψ occurring in ϕ . $\mathbf{N}_{\mathbb{C}}$ denotes a set of *constants*. If $\mathcal{I}, i \models \mathbf{a}(\phi)$, then \mathbf{a} is called an *answer* to ϕ w.r.t. \mathcal{I} at time

point i . The set of all answers to ϕ w.r.t \mathcal{I} at time point i is denoted by $\text{Ans}(\phi, \mathcal{I}, i)$.

As in [?], we can show that

- $\Box\phi_1$ is equivalent to $\phi_1 \wedge \bullet\Box\phi_1$;
- $\Diamond\phi_1$ is equivalent to $\phi_1 \vee \bigcirc\Diamond\phi_1$.

Because of the way \Box is defined in [?] we have to use \bullet instead of \bigcirc with the only difference that \bullet is tautological at the last time point. We can similarly show that

- $\Box^-\phi_1$ is equivalent to $\phi_1 \wedge \bullet^-\Box^-\phi_1$;
- $\Diamond^-\phi_1$ is equivalent to $\phi_1 \vee \bigcirc^-\Diamond^-\phi_1$.

Analogously to the reason as mentioned above we have to use \bullet^- instead of \bigcirc^- with the only difference that \bullet^- is tautological at the first time point. Thus at the last time point

- $\Box\phi_1$ is equivalent to ϕ_1 because $\bullet\Box\phi_1$ is tautological
- $\Diamond\phi_1$ is equivalent to ϕ_1 because $\bigcirc\Diamond\phi_1$ does not have any answers

and at the first time point

- $\Box^-\phi_1$ is equivalent to ϕ_1 because $\bullet^-\Box^-\phi_1$ is tautological
- $\Diamond^-\phi_1$ is equivalent to ϕ_1 because $\bigcirc^-\Diamond^-\phi_1$ does not have any answers

Proposition 0.2 (similar to Proposition 3.4 in [?]). *For $\mathfrak{a} : \text{FVar}(\phi) \rightarrow \mathbb{N}_C$ and $0 \leq i \leq n$, we have*

1. $\mathcal{I}, i \models \mathfrak{a}(\Box\phi_1)$ iff
 - $\mathcal{I}, i \models \mathfrak{a}(\phi_1)$ and
 - $i < n$ implies $\mathcal{I}, i + 1 \models \mathfrak{a}(\Box\phi_1)$
2. $\mathcal{I}, i \models \mathfrak{a}(\Box^-\phi_1)$ iff
 - $\mathcal{I}, i \models \mathfrak{a}(\phi_1)$ and
 - $i > 0$ implies $\mathcal{I}, i - 1 \models \mathfrak{a}(\Box^-\phi_1)$
3. $\mathcal{I}, i \models \mathfrak{a}(\Diamond\phi_1)$ iff

- $\mathfrak{I}, i \models \mathbf{a}(\phi_1)$ or
- $i < n$ and $\mathfrak{I}, i + 1 \models \mathbf{a}(\Diamond \phi_1)$

4. $\mathfrak{I}, i \models \mathbf{a}(\Diamond^- \phi_1)$ iff

- $\mathfrak{I}, i \models \mathbf{a}(\phi_1)$ or
- $i > 0$ and $\mathfrak{I}, i - 1 \models \mathbf{a}(\Diamond^- \phi_1)$

ANNOTATION: CASES 1 AND 4 MAY BE SUFFICIENT

Proof. To prove the above proposition we will prove each equivalence. We will mainly show this on the basis of the semantics.

1. $\Box \phi_1 \equiv \phi_1 \wedge \bullet \Box \phi_1$

$$\mathfrak{I}, i \models \mathbf{a}(\Box \phi_1) \quad (1)$$

$$\Leftrightarrow \mathfrak{I}, k \models \mathbf{a}(\phi_1) \text{ for all } k, i \leq k \leq n \quad (2)$$

$$\Leftrightarrow \mathfrak{I}, i \models \mathbf{a}(\phi_1) \text{ and } i < n \text{ implies } \mathfrak{I}, k \models \mathbf{a}(\phi_1) \text{ for all } k, i + 1 \leq k \leq n \quad (3)$$

$$\Leftrightarrow \mathfrak{I}, i \models \mathbf{a}(\phi_1) \text{ and } i < n \text{ implies } \mathfrak{I}, i + 1 \models \mathbf{a}(\Box \phi_1) \quad (4)$$

$$\Leftrightarrow \mathfrak{I}, i \models \mathbf{a}(\phi_1 \wedge \bullet \Box \phi_1) \quad (5)$$

(3) is equivalent to (2) because for $i < n$ in order to contain an answer the query needs to be satisfied now (at time point i) as well as at all future time points ($i + 1 \leq k \leq n$). Since $i < n$ is true the satisfaction of future time points solely depends on the second part of the "implies"-statement. In the case of $i = n$ this is equivalent as well because $\mathfrak{I}, i \models \mathbf{a}(\phi_1) \Leftrightarrow \mathfrak{I}, k \models \mathbf{a}(\phi_1)$ for all $k, n \leq k \leq n$ and $i < n$ is not true.

2. $\Box^- \phi_1 \equiv \phi_1 \wedge \bullet^- \Box^- \phi_1$

$$\mathfrak{I}, i \models \mathbf{a}(\Box^- \phi_1) \quad (6)$$

$$\Leftrightarrow \mathfrak{I}, k \models \mathbf{a}(\phi_1) \text{ for all } k, 0 \leq k \leq i \quad (7)$$

$$\Leftrightarrow \mathfrak{I}, i \models \mathbf{a}(\phi_1) \text{ and } i > 0 \text{ implies } \mathfrak{I}, k \models \mathbf{a}(\phi_1) \text{ for all } k, 0 \leq k \leq i - 1 \quad (8)$$

$$\Leftrightarrow \mathfrak{I}, i \models \mathbf{a}(\phi_1) \text{ and } i > 0 \text{ implies } \mathfrak{I}, i - 1 \models \mathbf{a}(\Box^- \phi_1) \quad (9)$$

$$\Leftrightarrow \mathfrak{I}, i \models \mathbf{a}(\phi_1 \wedge \bullet^- \Box^- \phi_1) \quad (10)$$

(8) is equivalent to (7) because for $i > 0$ in order to contain an answer the query needs to be satisfied now (at time point i) as well as at all

past time points ($0 \leq k \leq i-1$). Since $i > 0$ is true the satisfaction of past time points solely depends on the second part of the "implies"-statement. In the case of $i = 0$ this is equivalent as well because $\mathcal{I}, i \models \mathbf{a}(\phi_1) \Leftrightarrow \mathcal{I}, k \models \mathbf{a}(\phi_1)$ for all $k, 0 \leq k \leq 0$ and $i > 0$ is not true.

$$3. \Diamond\phi_1 \equiv \phi_1 \vee \bigcirc\Diamond\phi_1$$

$$\mathcal{I}, i \models \mathbf{a}(\Diamond\phi_1) \quad (11)$$

$$\Leftrightarrow \mathcal{I}, k \models \mathbf{a}(\phi_1) \text{ for some } k, i \leq k \leq n \quad (12)$$

$$\Leftrightarrow \mathcal{I}, i \models \mathbf{a}(\phi_1) \text{ or } i < n \text{ and } \mathcal{I}, k \models \mathbf{a}(\phi_1) \text{ for some } k, i+1 \leq k \leq n \quad (13)$$

$$\Leftrightarrow \mathcal{I}, i \models \mathbf{a}(\phi_1) \text{ or } i < n \text{ and } \mathcal{I}, i+1 \models \mathbf{a}(\Diamond\phi_1) \quad (14)$$

$$\Leftrightarrow \mathcal{I}, i \models \mathbf{a}(\phi_1 \vee \bigcirc\Diamond\phi_1) \quad (15)$$

(13) is equivalent to (12) because for $i < n$ in order to contain an answer the query needs to be satisfied now (at time point i) or at one future time point ($i+1 \leq k \leq n$). Since $i < n$ is true the satisfaction of future time points solely depends on the second part of the "implies"-statement. In the case of $i = n$ this is equivalent as well because $\mathcal{I}, i \models \mathbf{a}(\phi_1) \Leftrightarrow \mathcal{I}, k \models \mathbf{a}(\phi_1)$ for all $k, n \leq k \leq n$ and $i < n$ is not true but $\mathcal{I}, k \models \mathbf{a}(\phi_1)$ for some $k, n+1 \leq k \leq n$ does not hold either, so we do not lose any answers.

$$4. \Diamond^-\phi_1 \equiv \phi_1 \vee \bigcirc^-\Diamond^-\phi_1$$

$$\mathcal{I}, i \models \mathbf{a}(\Diamond^-\phi_1) \quad (16)$$

$$\Leftrightarrow \mathcal{I}, k \models \mathbf{a}(\phi_1) \text{ for some } k, 0 \leq k \leq i \quad (17)$$

$$\Leftrightarrow \mathcal{I}, i \models \mathbf{a}(\phi_1) \text{ or } i > 0 \text{ and } \mathcal{I}, k \models \mathbf{a}(\phi_1) \text{ for some } k, 0 \leq k \leq i-1 \quad (18)$$

$$\Leftrightarrow \mathcal{I}, i \models \mathbf{a}(\phi_1) \text{ or } i > 0 \text{ and } \mathcal{I}, i-1 \models \mathbf{a}(\Diamond^-\phi_1) \quad (19)$$

$$\Leftrightarrow \mathcal{I}, i \models \mathbf{a}(\phi_1 \vee \bigcirc^-\Diamond^-\phi_1) \quad (20)$$

(18) is equivalent to (17) because for $i > 0$ in order to contain an answer the query needs to be satisfied now (at time point i) or at any past time point ($0 \leq k \leq i-1$). Since $i > 0$ is true the satisfaction of past time points solely depends on the second part of the "implies"-statement. In the case of $i = 0$ this is equivalent as well because $\mathcal{I}, i \models \mathbf{a}(\phi_1) \Leftrightarrow \mathcal{I}, k \models \mathbf{a}(\phi_1)$ for some $k, 0 \leq k \leq 0$ and $i > 0$ is not true but $\mathcal{I}, k \models \mathbf{a}(\phi_1)$ for some $k, 0 \leq k \leq 0-1$ does not hold either, so we do not lose any answers.

□

The semantics of the four queries can be used now to extend the algorithm given in [?]. For that we need the notation of *answer terms*. We use the same simplification and assume in the following that N_V , the set of *variables*, is finite and that answers are of the form $\mathbf{a} : N_V \rightarrow \Delta$ instead of $\mathbf{a} : FVar(\phi) \rightarrow \Delta$. $Ans(\phi, \mathcal{J}^{(n)})$ refers to a set of mappings $\mathbf{a} : N_V \rightarrow \Delta$, i.e, a subset of Δ^{N_V} .

Definition 0.3 (answer term cf. Definition 6.1 in [?]). Let $FSub(\phi)$ denote the set of all subqueries of ϕ of the form $\circ\psi_1, \bullet\psi_1, \square\psi_1, \diamond\psi_1$ or $\psi_1 \cup \psi_2$. For $j \geq 0$, we denote by Var_j^ϕ the set of all variables of the form x_j^ψ for $\psi \in FSub(\phi)$. The set AT_ϕ^i of all *answer terms* for ϕ at $i \geq 0$ is the smallest set satisfying the following conditions:

- Every set $A \subseteq \Delta^{N_V}$ is an answer term for ϕ at i .
- Every variable $x_j^\psi \in Var_j^\phi$ with $j \leq i$ is an answer term for ϕ at i .
- If α_1 and α_2 are answer terms for ϕ at i , then so are $\alpha_1 \cap \alpha_2$ and $\alpha_1 \cup \alpha_2$.

The functions $eval^n : AT_\phi^n \rightarrow 2^{\Delta^{N_V}}$, $n \geq 0$ in [?] have then to be extended as follows:

α	$eval^n(\alpha)$
$x_j^{\square\psi_1}$ with $j < n$	$Ans(\square\psi_1, \mathcal{J}^{(n)}, j+1)$
$x_j^{\diamond\psi_1}$ with $j < n$	$Ans(\diamond\psi_1, \mathcal{J}^{(n)}, j+1)$
$x_n^{\square\psi_1}$	Δ^{N_V}
$x_n^{\diamond\psi_1}$	\emptyset

Table 2: $eval^n(\alpha)$

The function $\Phi_0(\psi) : Sub(\phi) \rightarrow AT_\phi^0$ in [?] has to be expanded as follows:

ψ	$\Phi_0(\psi)$
$\Box\psi_1$	$\Phi_0(\psi_1) \cap x_0^{\Box\psi_1}$
$\Box^-\psi_1$	$\Phi_0(\psi_1)$
$\Diamond\psi_1$	$\Phi_0(\psi_1) \cup x_0^{\Diamond\psi_1}$
$\Diamond^-\psi_1$	$\Phi_0(\psi_1)$

Table 3: $\Phi_0(\psi)$

The function $\Phi_i^0(\psi) : \text{Sub}(\phi) \rightarrow \text{AT}_\phi^i$, $i > 0$ in [?] has to be extended as follows:

ψ	$\Phi_i^0(\psi)$
$\Box\psi_1$	$\Phi_i^0(\psi_1) \cap x_i^{\Box\psi_1}$
$\Box^-\psi_1$	$\Phi_i^0(\psi_1) \cap \Phi_{i-1}(\Box^-\psi_1)$
$\Diamond\psi_1$	$\Phi_i^0(\psi_1) \cup x_i^{\Diamond\psi_1}$
$\Diamond^-\psi_1$	$\Phi_i^0(\psi_1) \cup \Phi_{i-1}(\Diamond^-\psi_1)$

Table 4: $\Phi_i^0(\psi)$

$\text{Sub}(\phi)$ denotes the set of all TQs occurring as temporal subqueries in ϕ (including ϕ itself).

To prove that correctness and boundedness of the algorithm is preserved we will add the necessary cases to the corresponding proofs from [?].

Lemma 0.4 (cf. Lemma 6.3 in [?]). *The function Φ_0 is correct for 0.*

Proof. We show by induction on the structure of the subqueries $\psi \in \text{Sub}(\phi)$ that $\text{eval}^n(\Phi_0(\psi))$ is equal to $\text{Ans}(\psi, \mathcal{J}^{(n)}, 0)$ for all $n \geq 0$.

If $\psi = \Box^-\psi_1$ or $\psi = \Diamond^-\psi_1$ then

$$\text{eval}^n(\Phi_0(\psi)) = \text{eval}^n(\Phi_0(\psi_1)).$$

This is by induction equal to $\text{Ans}(\psi_1, \mathcal{J}^{(n)}, 0)$ which then is, as shown in Proposition 0.2, equal to $\text{Ans}(\psi, \mathcal{J}^{(n)}, 0)$.

If $\psi = \Box\psi_1$, then

$$\begin{aligned}
\text{eval}^n(\Phi_0(\psi)) &= \text{eval}^n(\Phi_0(\psi_1)) \cap \text{eval}^n(x_0^\psi) \\
&= \text{Ans}(\psi_1, \mathcal{J}^{(n)}, 0) \cap \left\{ \begin{array}{ll} \text{Ans}(\psi, \mathcal{J}^{(n)}, 1) & \text{if } n > 0 \\ \Delta^{\mathbf{N}_v} & \text{if } n = 0 \end{array} \right\} \\
&= \text{Ans}(\psi, \mathcal{J}^{(n)}, 0)
\end{aligned}$$

If $\psi = \diamond\psi_1$, then

$$\begin{aligned}\text{eval}^n(\Phi_0(\psi)) &= \text{eval}^n(\Phi_0(\psi_1)) \cup \text{eval}^n(x_0^\psi) \\ &= \text{Ans}(\psi_1, \mathfrak{I}^{(n)}, 0) \cup \begin{cases} \text{Ans}(\psi, \mathfrak{I}^{(n)}, 1) & \text{if } n > 0 \\ \emptyset & \text{if } n = 0 \end{cases} \\ &= \text{Ans}(\psi, \mathfrak{I}^{(n)}, 0)\end{aligned}$$

□

Lemma 0.5 (cf. Lemma 6.4 in [?]). *If Φ_{i-1} is correct for $i-1$, then Φ_i^0 is correct for i .*

Proof. We show by induction on the structure of the subqueries $\psi \in \text{Sub}(\phi)$ that $\text{eval}^n(\Phi_i^0(\psi))$ is equal to $\text{Ans}(\psi, \mathfrak{I}^{(n)}, i)$ for all $n \geq i$.

If $\psi = \square^-\psi_1$, then

$$\begin{aligned}\text{eval}^n(\Phi_i^0(\psi)) &= \text{eval}^n(\Phi_i^0(\psi_1)) \cap \text{eval}^n(\Phi_{i-1}(\psi)) \\ &= \text{Ans}(\psi_1, \mathfrak{I}^{(n)}, i) \cap \text{Ans}(\psi, \mathfrak{I}^{(n)}, i-1) \\ &= \text{Ans}(\psi, \mathfrak{I}^{(n)}, i)\end{aligned}$$

If $\psi = \diamond^-\psi_1$, then

$$\begin{aligned}\text{eval}^n(\Phi_i^0(\psi)) &= \text{eval}^n(\Phi_i^0(\psi_1)) \cup \text{eval}^n(\Phi_{i-1}(\psi)) \\ &= \text{Ans}(\psi_1, \mathfrak{I}^{(n)}, i) \cup \text{Ans}(\psi, \mathfrak{I}^{(n)}, i-1) \\ &= \text{Ans}(\psi, \mathfrak{I}^{(n)}, i)\end{aligned}$$

If $\psi = \square\psi_1$, then

$$\begin{aligned}\text{eval}^n(\Phi_i^0(\psi)) &= \text{eval}^n(\Phi_i^0(\psi_1)) \cap \text{eval}^n(x_i^\psi) \\ &= \text{Ans}(\psi_1, \mathfrak{I}^{(n)}, i) \cap \begin{cases} \text{Ans}(\psi, \mathfrak{I}^{(n)}, i+1) & \text{if } n > i \\ \Delta^{\text{Nv}} & \text{if } n = i \end{cases} \\ &= \text{Ans}(\psi, \mathfrak{I}^{(n)}, i)\end{aligned}$$

If $\psi = \diamond\psi_1$, then

$$\begin{aligned}\text{eval}^n(\Phi_i^0(\psi)) &= \text{eval}^n(\Phi_i^0(\psi_1)) \cup \text{eval}^n(x_i^\psi) \\ &= \text{Ans}(\psi_1, \mathfrak{I}^{(n)}, i) \cup \begin{cases} \text{Ans}(\psi, \mathfrak{I}^{(n)}, i+1) & \text{if } n > i \\ \emptyset & \text{if } n = i \end{cases} \\ &= \text{Ans}(\psi, \mathfrak{I}^{(n)}, i)\end{aligned}$$

□

Lemma 0.6 (cf. Lemma 6.5 in [?]). *If Φ_{i-1} is correct for $i-1$ and $(i-1)$ -bounded, then we can construct a function $\Phi_i : \text{Sub}(\phi) \rightarrow \text{AT}_\phi^i$ that is correct for i and i -bounded.*

Proof. We need to expand the introduced function $\text{update}(x_{i-1}^{\psi^j})$ before then showing for all $n \geq i$ that $\text{eval}^n(x_{i-1}^{\psi^j})$ is still equal to $\text{eval}^n(\text{update}(x_{i-1}^{\psi^j}))$. After considering the new operators $\text{update}(x_{i-1}^{\psi^j})$ looks like this:

$$\text{update}(x_{i-1}^{\psi^j}) := \left\{ \begin{array}{ll} \Phi_i^{j-1}(\psi_1) & \text{if } \psi^j = \circ\psi_1 \text{ or } \psi^j = \bullet\psi_1 \\ \Phi_i^{j-1}(\psi^j) & \text{if } \psi^j = \psi_1 U \psi_2 \text{ or } \psi^j = \square\psi_1 \text{ or } \psi^j = \diamond\psi_1 \end{array} \right\}$$

For $\psi^j = \square\psi_1$ and $\psi^j = \diamond\psi_1$, by definition $\text{eval}^n(x_{i-1}^{\psi^j}) = \text{Ans}(\psi^j, \mathcal{J}^{(n)}, i)$.

Since Φ_i^{j-1} is correct for i , this is the same set as $\text{eval}^n(\Phi_i^{j-1}(\psi^j)) = \text{eval}^n(\text{update}(x_{i-1}^{\psi^j}))$.

It remains to show i -boundedness of $\Phi_i = \Phi_i^k$. In [?] this is again proven by induction on j . It therefore suffices to add the missing cases. It is enough to show that $\text{update}(x_{i-1}^{\psi^j})$ contains only variables from $\text{Var}_i^{\psi^j}$. If $\psi^j = \square\psi_1$ or $\psi^j = \diamond\psi_1$, then $\text{update}(x_{i-1}^{\psi^j}) = \Phi_i^{j-1}(\psi^j)$. Since Φ_i^{j-1} differs from Φ_i^0 only in the replacement of some variables with index $i-1$

$$\Phi_i^{j-1}(\psi^j) = \Phi_i^{j-1}(\psi_1) \cap x_i^{\psi^j}$$

or

$$\Phi_i^{j-1}(\psi^j) = \Phi_i^{j-1}(\psi_1) \cup x_i^{\psi^j}, \text{ respectively.}$$

By the induction hypothesis $\Phi_i^{j-1}(\psi_1)$ contains only variables from $\text{Var}_i^{\psi_1} = \text{Var}_i^{\psi^j} \setminus \{x_i^{\psi^j}\}$ and $\text{Var}_{i-1}^{\psi_1} \cap \{x_{i-1}^{\psi^j}, \dots, x_{i-1}^{\psi^k}\}$. Since every variable $x_{i-1}^{\psi'}$ in $\text{Var}_{i-1}^{\psi_1}$ must satisfy $\psi' \in \text{FSub}(\psi_1)$ the second set $\text{Var}_{i-1}^{\psi_1} \cap \{x_{i-1}^{\psi^j}, \dots, x_{i-1}^{\psi^k}\}$ is empty. This follows from the total order $\psi^1 < \dots < \psi^k$ on the set $\text{FSub}(\phi) = \{\psi^1, \dots, \psi^k\}$ presented in [?], i.e., $\psi' \in \text{FSub}(\psi^j) \setminus \{\psi^j\}$, and thus $\psi' < \psi^j$. \square