# A Compact Representation for Least Common Subsumers in the Description Logic $\mathcal{ALE}$

Chan Le Duc [a], Nhan Le Thanh [a], Marie-Christine Rousset [b]
[a] *Laboratoire I3S, Université de Nice-Sophia Antipolis, France*
[b] *Laboratoire de Recherche en Informatique, Université Paris-Sud, France*

Abstract
This paper introduces a compact representation which helps to avoid the exponential blow-up of the Least Common Subsumer (*lcs*) of two $\mathcal{ALE}$-concept descriptions. Based on the compact representation we define a space of specific graphs which represents all $\mathcal{ALE}$-concept descriptions including the *lcs*. Next, we propose an algorithm exponential in time and polynomial in space for deciding subsumption between concept descriptions represented by graphs in this space. These results provide better understanding of the double exponential blow-up of the approximation of $\mathcal{ALC}$-concept descriptions by $\mathcal{ALE}$-concept descriptions: double exponential size of the approximation in the ordinary representation is inavoidable in the worst case.
Keywords: Description Logics, Least Common Subsumer, Approximation.

## 1. Introduction

Description logics can be used as a formalism for representing ontologies. The OWL language [10], which is becoming a standard language for ontologies, is founded on description logics. If we ignore role constructors and general concept inclusions, the OWL-Lite and OWL-DL [10] languages are respectively comparable to the $\mathcal{ALE}$ and $\mathcal{ALC}$ description logics. The major difference between $\mathcal{ALE}$ and $\mathcal{ALC}$ is that the disjunction constructor is absent from $\mathcal{ALE}$. As a result, deciding subsumption in $\mathcal{ALE}$ is NP-complete [6] while it is PSPACE-complete for $\mathcal{ALC}$ [7].
In this paper, we revisit two problems which have been recently addressed in description logics: the computation of the least common subsumer (*lcs*) of two concept descriptions and the approximation from a description logic $L_1$ to a description logic $L_2$.
As mentioned in recent work [3], computing the *lcs* is a useful inference task for the bottom-up construction of knowledge bases in description logics. It also can be used for computing similarity between concept description of different ontologies. Finally, it plays a central role for computing approximations. An algorithm for computing the approximation where $L_1 = \mathcal{ALC}$ and $L_2 = \mathcal{ALE}$ is presented in [1]. It returns a concept description whose size may be double exponential in the size of the input. This algorithm is based on an exponential algorithm which computes the *lcs* of concept descriptions in $\mathcal{ALE}$. As shown in [4], the exponential size of *lcs* cannot be avoided if we use the ordinary representation of *normalized* concept descriptions, whose size may be exponential compared to the initial (not normalized) concept descriptions.
Some recent results extend those presented in [3] and [1] to more expressive description logics. For instance a double exponential algorithm for computing *lcs* in $\mathcal{ALEN}$ is presented in [11]. It yields a double exponential upper bound for the size of the *lcs* of two $\mathcal{ALEN}$-concept descriptions. Another result described in [12] provides an algorithm for computing *lcs* in $\mathcal{FLE}+$ ($\mathcal{FLE}$ with transitive roles). Nevertheless, the complexity of this algorithm is not given.
The first contribution of this paper is a compact representation for $\mathcal{ALE}$-concept descriptions which avoids the exponential blow-up of the size of the description trees built from *normalized* concept descriptions as in [3]. This (polynomial) compact representation is a graph, which is directly built from the description trees obtained from the

AI Communications
ISSN 0921-7126, IOS Press. 

*weakly normalized* concept descriptions. A *weakly normalized* concept description is obtained by applying the normalization rules presented in [3] except for the normalization rule responsible of the exponential blow-up of the size of the normalized concept description (and thus of the resulting description tree). This new representation of a weakly normalized concept description is called its $\epsilon$-*tree* because it replaces the effective application of the expansive normalization rule by adding what we have called $\epsilon$-*edges* to the description trees corresponding to the *weakly normalized* concept descriptions. Then, *normalization graphs* are built from $\epsilon$-trees by making explicit bottom-concepts in labels of nodes in $\epsilon$-trees (as $\mathcal{ALE}$ allows for bottom-concept and negated concept names). We also obtain a polynomial compact representation of the *lcs* of two $\mathcal{ALE}$-concept descriptions by defining the *product* of two normalization graphs. Finally, we exploit this compact representation for providing an algorithm for checking subsumption between concept descriptions or *lcs* of concept descriptions in polynomial space and exponential time.

The second contribution of this paper is to show that the lower bound for computing the approximation of $\mathcal{ALC}$-concept descriptions by $\mathcal{ALE}$-concept descriptions is double exponential in the size of the input. This result answers *partially*[1] the question left open in [1] on the existence of an exponential algorithm for the approximation of $\mathcal{ALC}$ by $\mathcal{ALE}$. It also shows the limit of the compact representation that we have introduced: it cannot prevent the exponential blow-up resulting from n-ary *lcs* computation.

The paper is organized as follows.

In Section 2, we provide the formal background on which this paper is based. In particular, we distinguish the weak normal form and the strong normal form of a concept description, depending on the normalization rules that are applied. We also recall the definition of description trees introduced in [3].

In Section 3, we define the $\epsilon$-tree of a weakly normalized concept description, and the normalization graph resulting from the $\epsilon$-tree. Next, we provide the transformation algorithm from a normalization graph into the description tree of the corresponding *normalized* concept description. We also provide a polynomial space (and exponential time) algorithm exploiting the normalization graphs for checking subsumption in $\mathcal{ALE}$.

In Section 4, we define the product of normalization graphs and we show how it can be exploited for computing a representation of the *lcs* of two $\mathcal{ALE}$-concept descriptions in polynomial time. We also show that the algorithms introduced in Section 2 for normalization graphs can be extended to their products.

In Section 5, we prove that the lower bound of the size of the approximation of an $\mathcal{ALC}$- concept description by an $\mathcal{ALE}$-concept description in the compact graph representation (and a fortiori in the ordinary representation) is double exponential in the size of the input. The reason is that the double exponential blow-up is due to the computation of the n-ary *lcs*, and the compact graph representation that we have introduced in this paper cannot prevent the exponential blow-up resulting from the computation of the n-ary *lcs*.

Finally, Section 6 concludes and provides a brief discussion on the results obtained in this paper.

## 2. Formal background

In this section we will briefly present important notions of description logics and existing results about the *lcs* computation. Details of this formalism can be found in [9]. Let $N_C$ be a set of primitive concepts and $N_R$ be a set of primitive roles. The logic $\mathcal{ALE}$ uses the following constructors to build concept descriptions : conjunction ($C \sqcap D$), value restriction ($\forall r.C$), existential restriction ($\exists r.C$), primitive negation ($\neg P$), top-concept and bottom-concept. The logic $\mathcal{ALC}$ uses all of these constructors together with disjunction ($C \sqcup D$).

Let $\Delta$ be a non-empty set of individuals. Let $\cdot^{\mathcal{I}}$ be a function that transforms each primitive concept $P \in N_C$ into $P^I \subseteq \Delta$ and each primitive role $r \in N_R$ into $r^I \subseteq \Delta \times \Delta$. The semantics of a concept description are inductively defined owing to the interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ as in the table below.

**Subsumption** Let $C$, $D$ be concept descriptions. $D$ subsumes $C$, $C \sqsubseteq D$, iff $C^I \subseteq D^I$ for all interpretations $\mathcal{I}$.

**Least Common Subsumer** Let $C_1$, $C_2$ be concept descriptions in a DL. $C$ is a least common subsumer of $C_1$, $C_2$ ($lcs(C_1, C_2)$ for short) iff $C_i \sqsubseteq C$

[1]This will be discussed in Section 8 of this paper.

| Syntax | Semantics |
|---|---|
| $\top$ | $\Delta$ |
| $\bot$ | $\emptyset$ |
| $C \sqcap D$ | $C^I \cap D^I$ |
| $C \sqcup D$ | $C^I \cup D^I$ |
| $\forall r.C$, $r \in N_R$ | $\{x \in \Delta \mid \forall y{:}(x,y) \in r^I \to y \in C^I\}$ |
| $\neg C$ | $\Delta \setminus C^I$ |
| $\exists r.C$, $r \in N_R$ | $\{x \in \Delta \mid \exists y{:}(x,y) \in r^I \wedge y \in C^I\}$ |

for all $i$ , and if $C'$ is a concept description such that $C_i \sqsubseteq C'$ for all $i$, then $C \sqsubseteq C'$.

**Approximation** Let $C$ be a concept description in a $L_1$ and $D$ be a concept description in a $L_2$ where $L_1, L_2$ are DLs. $D$ is called upper $L_2$-approximation of $C$ ($D = approx_{\mathcal{L}_1}(C)$ for short) iff $C \sqsubseteq D$ and, if $C \sqsubseteq D'$ and $D' \sqsubseteq D$, then $D' \equiv D$ for all $L_2$-concept description $D'$.

In what follows, we will recall important notions and results in [3] on which the work of this paper is founded.

The depth of an $\mathcal{ALE}$-concept description $C$ is inductively defined as follows: i) $depth(P) = depth(\neg P) = depth(\top) = depth(\bot) := 0$ ($P \in N_C$); ii) $depth(C \sqcap D) := max(depth(C), depth(D))$; iii) $depth(\forall r.C) = depth(\exists r.C) := depth(C) + 1$.

**Definition 1** ($\mathcal{ALE}$*-description tree*) [3] *Given a set $N_C$ of primitive concepts and a set $N_R$ of primitive roles, a description tree is of the form $\mathcal{G}$ = ($V$, $E$, $v^0$, $l$) where*

- *$V$ is the set of nodes of $\mathcal{G}$;*
- *$E \subseteq V \times (N_R \cup \forall N_R) \times V$ is a finite set of edges labeled with role names $r$ ($\exists$-edges) or with $\forall r$ ($\forall$-edges); $\forall N_R := \{\forall r \mid r \in N_R\}$;*
- *$v^0$ is the root of $\mathcal{G}$;*
- *$l$ is a labeling function mapping the nodes in $V$ to finite set $\{P_1, ..., P_k\}$ where each $P_i$, $1 \le i \le k$, is one of the following forms : $P_i \in N_C$, $P_i = \neg P$ for some $P \in N_C$, or $P_i = \bot$. The empty label corresponds to the top-concept $\top$.*

In [3], the authors have proposed a procedure for transforming an $\mathcal{ALE}$-concept description $C$ into the corresponding $\mathcal{ALE}$-description tree $\mathcal{G}(C)$ = ($V$, $E$, $v^0$, $l$) as follows. Every $\mathcal{ALE}$-concept description $C$ can be written as $C \equiv P_1 \sqcap ... \sqcap P_n \sqcap \exists r_1 C_1 \sqcap ... \sqcap \exists r_m C_m \sqcap \exists s_1 D_1 \sqcap ... \sqcap \exists s_k D_k$ where $P_i \in N_C \cup \{\neg P_i \mid P_i \in N_C\} \cup \{\top, \bot\}$. Then,

If $depth(C) = 0$ then $V := \{v^0\}$, $E := \emptyset$ and $l(v^0) := \{P_1, ..., P_n\} \setminus \{\top\}$.

If $depth(C) > 0$ then for $1 \le i \le m$, let $\mathcal{G}_i = (V_i, E_i, v_i^0, l_i)$ be the inductively defined $\mathcal{ALE}$-description tree corresponding to $C_i$, and for $1 \le j \le k$, let $\mathcal{G}'_j = (V'_j, E'_j, v'^0_j, l'_j)$ be the inductively defined $\mathcal{ALE}$-description tree corresponding to $D_j$ where $V_i$ and $V'_j$ are pairwise disjoint. Then,

- $V := \{v^0\} \cup V_i \cup V'_j$,
- $E := \{(v^0 r_i v_i^0) \mid 1 \le i \le m\} \cup \{(v^0 \forall s_j v'^0_j) \mid 1 \le j \le k\} \cup \bigcup_{1 \le i \le m} E_i \cup \bigcup_{1 \le j \le k} E'_j$,
- $l(v) := \begin{cases} \{P_1, ..., P_n\} \setminus \{\top\}, v = v^0 \\ l_i(v), & v \in V_i, 1 \le i \le m \\ l'_j(v), & v \in V'_j, 1 \le j \le k \end{cases}$

Conversely, every $\mathcal{ALE}$-description tree $\mathcal{G}$ = ($V$, $E$, $v^0$, $l$) can be transformed into an $\mathcal{ALE}$-concept description $C_{\mathcal{G}}$ as follows.

If $depth(\mathcal{G}) = 0$ then $V = \{v^0\}$ and $E = \emptyset$. If $l(v^0) = \emptyset$ then $C_{\mathcal{G}} = \top$, otherwise, we have $l(v^0) = \{P_1, ..., P_n\}$, $n \ge 1$ , $P_i \in N_C \cup \{\bot\}$ and define $C_{\mathcal{G}} := P_1 \sqcap ... \sqcap P_n$.

If $depth(\mathcal{G}) > 0$ then $l(v^0) = \{P_1, ..., P_n\}$, $n \ge 0$ , $P_i \in N_C \cup \{\neg P_i \mid P_i \in N_C\} \cup \{\bot\}$ and let $\{v_1, ..., v_m\}$ be the set of all successors of $v^0$ where $(v^0 r_i v_i) \in E$, $1 \le i \le m$ for some $r_i \in N_R$, and let $\{w_1, ..., w_k\}$ be the set of all successors of $w^0$ where $(v^0 \forall s_i w_i) \in E$, $1 \le i \le k$ for some $s_i \in N_R$. Furthermore, let $C_1, ..., C_m$ ($D_1, ..., D_k$) the inductively defined $\mathcal{ALE}$-concept descriptions corresponding to the subtrees of $\mathcal{G}$ with roots $v_i$, $1 \le i \le m$ ($w_i$, $1 \le i \le k$). We define $C_{\mathcal{G}} := P_1 \sqcap ... \sqcap P_n \sqcap \exists r_1.C_1 \sqcap ... \sqcap \exists r_m.C_m \sqcap \forall s_1.D_1 \sqcap ... \sqcap \forall s_k D_k$.

The definition of the depth for $\mathcal{ALE}$-concept descriptions corresponds to the depth of its description tree. In addition, a node $v \in V$ of a description tree is called $\forall r$-successor ($r$-successor) if there exists an edge $(w \forall r v) \in E$ ($(wrv) \in E$). In this case, we also say that $v$ is a $\forall r$-successor ($r$-successor) of $w$.

**Definition 2** (*normalization rules*) [3] *The normal form of an $\mathcal{ALE}$-concept description $C$ is obtained from $C$ by exhaustively applying the following normalization rules:*

1. $\forall r.E \sqcap \forall r.F \to \forall r.(E \sqcap F)$
2. $\forall r.E \sqcap \exists r.F \to \forall r.\ E \sqcap \exists r.(E \sqcap F)$
3. $\forall r.\top \to \top$
4. $E \sqcap \top \to E$

5. $P \sqcap \neg P \rightarrow \bot$ for each $P \in N_C$
6. $\exists r.\bot \rightarrow \bot$
7. $E \sqcap \bot \rightarrow \bot$

*where E, F are two $\mathcal{ALE}$-concept descriptions and $r \in N_R$.*

Note that rules 3, 4 as specified in Definition 2 need to be applied once to $\mathcal{ALE}$-concept descriptions. However, the application of rule 2 (rule 1) can lead to the application of rules 1 (rule 2), 5, 6, 7 more times again. The normalization of an $\mathcal{ALE}$-concept description $C$ can be carried out in two steps. The first step consists of the application of all rules as specified in Definition 2 except for rule 2. This step yields an $\mathcal{ALE}$-concept description $C'$ where $C' \equiv C$, whose each conjunction contains at most one value restriction (at the same depth as the conjunction). The second step consists of the application of rules 1, 2, 5, 6, 7 to the concept description $C'$. In the second step, rules 1, 2 need to be exhaustively applied once since the application of rules 5, 6, 7 does not lead to the application of rules 1, 2 again. The concept description obtained from the second step is in the normal form according to Definition 2.
From these remarks, we introduce *weak* and *strong* normal forms for $\mathcal{ALE}$-concept descriptions, corresponding to the two normalization steps described above.

**Definition 3** *An $\mathcal{ALE}$-concept description $C$ is in* weak normal form *if $C$ is obtained from an $\mathcal{ALE}$-concept description by exhaustively applying all rules as specified in Definition 2, with the exception of rule 2.. Additionally, an $\mathcal{ALE}$-concept description $C'$ is in* strong normal form *if $C'$ is obtained from an $\mathcal{ALE}$-concept description in* weak normal form *by applying exhaustively rules 1, 2, 5, 6, 7 as specified in Definition 2.*

It is obvious that the application of all rules (as specified in Definition 2) with the exception of rule 2, does not increase the size of concept descriptions. However, as shown in [4], the size of $\mathcal{ALE}$-concept descriptions in strong normal form may increase exponentially. This exponential blow-up in space is caused by the application of rule 2. Example 1, which is taken from [4], demonstrates this effect.

**Example 1** *We define the following sequence $C_1$, $C_2$, $C_3$, ... of $\mathcal{ALE}$-concept descriptions*

$$C_n := \begin{cases} \exists r.P \sqcap \exists r.Q & n = 1 \\ \exists r.P \sqcap \exists r.Q \sqcap \forall r.C_{n-1}, & n > 1 \end{cases}$$

*At each level of the tree $\mathcal{G}(C_n)$, the application of rule 2. leads to copy from subtrees of $\forall r$-successors to $r$-successors. This implies that for each level of the tree $\mathcal{G}(C_n)$, the application of rule 2. generates two new $r$-successors for each $r$-successor. Therefore, the tree obtained from $\mathcal{G}(C_n)$ by applying rule 2 has at least $2^n$ nodes at level $n$.*

Accordingly with the notation introduced in [3], we denote $\mathcal{G}_C$ the description tree obtained from a concept description $C$ in *strong normal form*, and we denote $\mathcal{G}(C)$ the description tree obtained from a concept description $C$ in *weak normal form*. We transfer the semantics of concept descriptions to description trees as follows: for an interpretation $(\Delta, .^{\mathcal{I}})$, and a concept description $C$ in strong (respectively weak) normal form: $\mathcal{G}_C^{\mathcal{I}} := C^{\mathcal{I}}$ (respectively $\mathcal{G}(C)^{\mathcal{I}} := C^{\mathcal{I}}$). Note that $C \equiv C_{\mathcal{G}_C}$ and $C \equiv C_{\mathcal{G}(C)}$ since the normalization rules preserve equivalence.
In addition, let $\mathcal{G} = (V, E, v^0, l)$ be an $\mathcal{ALE}$-description tree. We denote $\mathcal{G}(v_i) = (V_{\mathcal{G}(v_i)}, V_{\mathcal{G}(v_i)}, v_i, l_{\mathcal{G}(v_i)})$ as the subtree of $\mathcal{G}$ whose root is $v_i \in V$.

**Definition 4** (*homomorphism*) [3] *A mapping $\varphi : V_H \rightarrow V_G$ from an $\mathcal{ALE}$-description tree $\mathcal{H} = (V_H, E_H, m_0, l_H)$ to an $\mathcal{ALE}$-description tree $\mathcal{G} = (V_G, E_G, n_0, l_G)$ is called homomorphism iff, the following conditions are satisfied:*

1. $\varphi(m_0) = n_0$
2. *For all $n \in V_H$, we have $l_H(n) \subseteq l_G(\varphi(n))$ or $l_G(\varphi(n)) = \{\bot\}$ ;*
3. *For all $n$r$m \in E_H$ , either $\varphi(n)\mathrm{r}\varphi(m) \in E_G$, or $\varphi(n) = \varphi(m)$ and $l_G(\varphi(n)) = \{\bot\}$; and*
4. *For all $n\forall rm \in E_H$, either $\varphi(n)\forall r\varphi(m) \in E_G$, or $\varphi(n) = \varphi(m)$ and $l_G(\varphi(n)) = \{\bot\}$.*

Additionally, if $\varphi$ is a bijection and $\varphi^{-1}$ is also a homomorphism from $\mathcal{G}$ to $\mathcal{H}$, then $\varphi$ is called *isomorphism*.
Note that the existence of two homomorphisms: $\varphi$ from $\mathcal{H}$ to $\mathcal{G}$ and $\varphi'$ from $\mathcal{G}$ to $\mathcal{H}$, does not imply that there exists an isomorphism from$\mathcal{H}$ to $\mathcal{G}$. In general, it is not necessary that $\varphi'$ is a bijection of $\varphi$.

A polynomial algorithm for checking the existence of a homomorphism between two $\mathcal{ALE}$-description trees has been proposed in [3]. Moreover, the authors have shown that the characterization of subsumption by homomorphisms requires that description trees must be built from $\mathcal{ALE}$-concept descriptions in strong normal form.

**Theorem 1** [3] *Let C, D be $\mathcal{ALE}$-concept descriptions, then $C \sqsubseteq D$ if and only if there exists a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$.*

Therefore, if we use *directly* the algorithm in [3] for checking whether there exists a homomorphism between such two $\mathcal{ALE}$-description trees $\mathcal{G}_C$ and $\mathcal{G}_D$, it will take an exponential space in the worst case since the size of these trees may be exponential in the size of input concept descriptions.
According to the work in [3], there always exists the *lcs* of $\mathcal{ALE}$-concept descriptions and it is unique. The computing of the *lcs* for two $\mathcal{ALE}$-concept descriptions $C, D$ requires that $C, D$ are in strong normal form. Next, the normalized concept descriptions have to be transformed into $\mathcal{ALE}$-description trees $\mathcal{G}_C$ and $\mathcal{G}_D$. The $\mathcal{ALE}$-description tree for the *lcs* will be the product tree of trees $\mathcal{G}_C$ and $\mathcal{G}_D$.

## 3. $\mathcal{E}$-trees and normalization graphs

This section is aimed at introducing a specific data structure, called *normalization graph*, for representing strong normal $\mathcal{ALE}$-concept descriptions in polynomial space.
The section will begin by introducing $\epsilon$-trees, denoted as $\mathcal{G}^\epsilon(C)$, which are built from description trees corresponding to $\mathcal{ALE}$-concept descriptions in weak normal form. This structure allows for substituting the application of rules 1, 2 (as specified in Definition 2) to $\mathcal{ALE}$-concept descriptions by adding $\epsilon$-edges to the corresponding description trees. Normalization graphs, denoted as $\mathcal{G}^\epsilon_C$, are formed from $\epsilon$-trees by adding some elements in order to capture rules 5, 6, 7 (as specified in Definition 2). Next, an algorithm for transforming normalization graphs into $\mathcal{ALE}$-description trees will be presented. We will show that description trees obtained from normalization graphs by applying this algorithm are isomorph to description trees built from $\mathcal{ALE}$-concept descriptions in strong normal form. The section will be terminated by an algorithm for deciding subsumption between concept descriptions represented by normalization graphs.
We need the following notations for the section. We denote $N'_C$ as the union $N_C \cup \{\neg P \mid P \in N_C\} \cup \{\bot\}$. Let $\mathcal{G} = (V_G,\ E_G,\ v^0,\ l_G)$ be an $\mathcal{ALE}$-description tree. We denote $|\mathcal{G}|$ as the depth of $\mathcal{G}$ and $v^k$ as a node at level $k$ of $\mathcal{G}$ where $v^k \in V_G$. Hence, we can write $(v^k e v^{k+1}) \in E_G$ for all $0 \leq k \leq |\mathcal{G}|$.
For the sake of simplicity, we can assume that $N_R = \{r\}$. All result obtained can be applied to a general set $N_R$.

**Definition 5** ($\epsilon$-*tree*) *Let $C$ be an $\mathcal{ALE}$-concept description in weak normal form and $\mathcal{G}(C) = (V, E, v^0, l)$ be its description tree. The $\epsilon$-tree $\mathcal{G}^\epsilon(C) = (V, E \cup E^\epsilon, l)$ is built from $\mathcal{G}(C)$ as follows:*

1. *For each $v \in V$, an $\epsilon$-edge $(v\epsilon v)$ is added to $E^\epsilon$.*
2. *For each level $k$ where $0 \leq k \leq |\mathcal{G}| - 1$, for each $\epsilon$-edge $(v_i^k \epsilon v_j^k)$ at level $k$ where $v_i^k \neq v_j^k$,*
   (a) *If there exist two edges $(v_i^k \forall r v_i^{k+1})$, $(v_j^k \forall r v_j^{k+1}) \in E$, then the $\epsilon$-edge $(v_i^{k+1} \epsilon v_j^{k+1})$ is added to $E^\epsilon$.*
   (b) *If there exist two edges $(v_i^k r v_i^{k+1})$, $(v_j^k \forall r v_j^{k+1}) \in E$, then the $\epsilon$-edge $(v_i^{k+1} \epsilon v_j^{k+1})$ is added to $E^\epsilon$.*
   (c) *If there exist two edges $(v_j^k r v_j^{k+1})$, $(v_i^k \forall r v_i^{k+1})$, then the $\epsilon$-edge $(v_j^{k+1} \epsilon v_i^{k+1})$ is added to $E^\epsilon$.*
3. *Node $v^0$ is called the root of the $\epsilon$-tree $\mathcal{G}^\epsilon(C)$. For each node $v \in V$, its predecessor in $\mathcal{G}^\epsilon(C)$, denoted as $p(v)$, is its predecessor in $\mathcal{G}(C)$. The level of a node $v \in V$ in $\mathcal{G}^\epsilon(C)$ is defined as being the depth of the node $v$ in $\mathcal{G}(C)$.*

Note that $\mathcal{G}^\epsilon(C)$ as defined in Definition 5 is an oriented graph. However, it becomes a tree if the $\epsilon$-edges are deleted from that graph. Let $(vev') \in E$ where $e \in N_R \cup \{\forall r | r \in N_R\}$. We say that $v'$ is an $e$-successor of $v$, or $v$ is an $e$-predecessor of $v'$. If $(v\epsilon v') \in E^\epsilon$, we say that $v'$ is an $\epsilon$-successor of $v$. We denote $p(p(...p(v)...)$ ($n$ times) as $p^n(v)$, $v \in V$ and $p^0(v) = v$.

**Remark 1** *The transformation of an $\mathcal{ALE}$-concept description $C$ into the $\epsilon$-tree as described in Defi-*

*nition 5 takes at most a polynomial time in the size of $C$. In fact, it holds that the size of the weak normal form of $C$ is bounded by the size of $C$ and the number of added $\epsilon$-edges is bounded by $|V|$ where $|V|$ is the number of nodes of the description tree obtained from the weak normal form of $C$.*

According to the work in [3], if $\mathcal{ALE}$-concept descriptions are represented by $\mathcal{ALE}$-description trees, the normalization by the rules in Definition 2 leads to copy $\forall r$-subtrees to $r$-successors in $\mathcal{ALE}$-description trees. The aim of $\epsilon$-trees is to avoid the copying of subtrees by memorizing "references" to subtrees to be copied. These references are represented as $\epsilon$-edges in $\epsilon$-trees. However, a $\forall r$-subtree can be copied to many $r$-successors *i.e* one node may be connected to many nodes by $\epsilon$-edges. Hence, predecessor function $p$, involved in Definition 5, allows one to determine "right neighbours" of a node thanks to its predecessor. It means that a node at level $k$ may belong to many *k-neighbourhoods*, which is composed of right neighbour nodes. Definition 6 will formalize this idea.

**Definition 6** (*neighbourhood*) *Let $\mathcal{G}^\epsilon(C)$=$(V, E \cup E^\epsilon, l)$ be an $\epsilon$-tree where $v^0 \in V$ is its root. At level 0 of $\mathcal{G}^\epsilon(C)$, there is a unique* 0-neighbourhood*, denoted $N^0 = \{v^0\}$.*
*For each $(k-1)$-neighbourhood $n^{k-1} \in N^{k-1}$, $n^{k-1} = \{v_1^{k-1}, ..., v_m^{k-1}\} \subseteq V$ $(0 < k \leq |\mathcal{G}^\epsilon(C)|)$ such that $\perp \notin l(v_1^{k-1}) \cup ... \cup l(v_m^{k-1})$, the set $N^k(n^{k-1})$ of $k$-neighbourhoods generated from $n^{k-1}$ is defined as follows.*

1. *If there exists an edge $(v^{k-1}\forall r v^k) \in E$ such that $v^{k-1} \in n^{k-1}$ then we obtain a $k$-neighbourhood $n^k \in N^k(n^{k-1})$,*
$$n^k := \begin{cases} V^\epsilon(n^{k-1}), & V^\epsilon(n^{k-1}) \neq \emptyset \\ V^\forall(n^{k-1}), & V^\epsilon(n^{k-1}) = \emptyset \end{cases}$$
*where*
$V^\forall(n^{k-1})$:=$\{v^k | (v^{k-1}\forall r v^k) \in E, v^{k-1} \in n^{k-1}\}$
$V^\epsilon(n^{k-1})$:= $\{v_i^k | \ (v^k \epsilon v_i^k) \in E^\epsilon,\ v^k \in V^\forall(n^{k-1}),$
$v_i^k \notin V^\forall(n^{k-1}),\ p(v_i^k) \in n^{k-1}\}$
2. *For each $r$-successor $v^k$ of all $v_i^{k-1} \in n^{k-1}$, we obtain a $k$-neighbourhood $n^k \in N^k(n^{k-1})$,*
$n^k := \{v^k\} \cup V^\epsilon_{v^k}$ *where*
$V^\epsilon_{v^k}$:=$\{v_i^k | (v^k \epsilon v_i^k) \in E^\epsilon,\ p(v_i^k) \in n^{k-1}\}$

The *unique* $k$-neighbourhood $n^k \in N^k(n^{k-1})$ generated from $\forall r$-successors (as defined by item 1. in Definition 6), is called $\forall r$-neighbourhood of $n^{k-1}$. The $k$-neighbourhoods $n^k \in N^k(n^{k-1})$ generated from $r$-successors (as defined by item 2. in Definition 6), are called $r$-neighbourhoods of $n^{k-1}$. If there is not any confusion, we read $k$-neighbourhood, $\forall r$-neighbourhood and $r$-neighbourhood respectively for neighbourhood at level $k$, neighbourhood generated from $\forall r$-successors and neighbourhood generated from $r$-successors of nodes in a $(k-1)$-neighbourhood.

**Remark 2** *For $\epsilon$-trees $\mathcal{G}^\epsilon(C)$, the set of nodes $V^\epsilon(n^{k-1})$ as specified by item 1. in Definition 6 is always empty. Therefore, the $\forall r$-neighbourhood of a $(k-1)$-neighbourhood $n^{k-1}$ is determined by set $V^\forall(n^{k-1})$ i.e the $\forall r$-successors of all nodes in $n^{k-1}$. However, $V^\epsilon(n^{k-1})$ will play an important role for computing neighbourhoods in more complex graphs e.g normalization or product graphs (Sections 3 and 4).*

In the following, we denote $\mathsf{label}(n^k)$ as the label of a $k$-neighbourhood $n^k$= $\{v_1^k, ..., v_m^k\}$ where $\mathsf{label}(n^k) := l(v_1^{k-1}) \cup ... \cup l(v_m^{k-1})$ if $\perp \notin l(v_1^{k-1}) \cup ... \cup l(v_m^{k-1})$ and $\mathsf{label}(n^k) := \{\perp\}$, otherwise.

**Example 2** *Let $D := \exists r.(A \sqcap \forall r.C) \sqcap \exists r.(B \sqcap \forall r.B) \sqcap \forall r.(C \sqcap \exists r.A)$. The $\epsilon$-tree $\mathcal{G}^\epsilon(D)$ is illustrated in Figure 1.*

In this figure, each node is associated with its name, predecessor, and label. For example, node $v_1$ has predecessor $v_0$ and label $\{A\}$. Since there is an $\epsilon$-edge $(v_0 \epsilon v_0)$, $v_2$ is a $r$-successor of $v_0$ and $v_3$ is a $\forall r$-successor of $v_0$, hence the $\epsilon$-edge $(v_2 \epsilon v_3)$ is added according to the condition 2.(b) of Definition 5. In contrast, since there is not any $\epsilon$-edge between $v_1$ and $v_2$, no $\epsilon$-edge connects nodes $v_5$ and $v_6$. The value of predecessor function $p(v_i)$ is the $r$-predecessor or the $\forall r$-predecessor of $v_i$.
Intuitively, the neighbourhood notion allows us to determine nodes that have to be grouped when applying rules 1, 2. More precisely, computing the neighbourhoods of an $\epsilon$-tree $\mathcal{G}^\epsilon(C)$ yields the nodes of the description tree corresponding to the concept description $C'$ obtained from $C$ by applying exhaustively rules 1 and 2. The following algorithm performs this transformation i.e the algorithm transform a graph, in which the notions of level and neighbourhood are well defined, into an

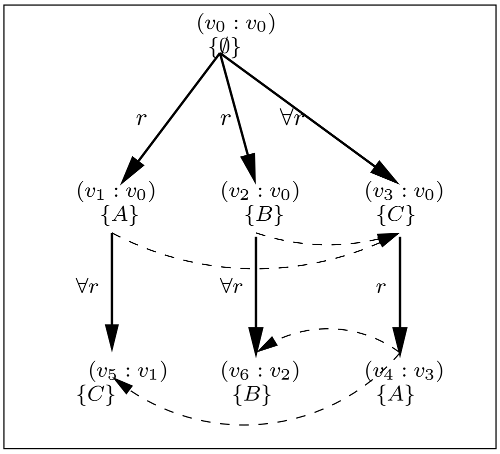


Figure 1. $\epsilon$-tree $\mathcal{G}^\epsilon(D)$

$\mathcal{ALE}$- description tree. To get started, we consider that the input graph of Algorithm 1 is an $\epsilon$-tree. Figure 2 illustrates the $\mathcal{ALE}$-description tree $\mathbf{B}(\mathcal{G}^\epsilon(D)) = (V', E', w_0, l')$ obtained from executing Algorithm 1 for the $\epsilon$-tree $\mathcal{G}^\epsilon(D) = (V, E \cup E^\epsilon, l)$ in Figure 1. At level 0, $\mathcal{G}^\epsilon(D)$ has only one 0-neighbourhood $(v_0)$. Thus, $\mathbf{B}(\mathcal{G}^\epsilon(D))$ has root $w_0$ where $l'(w_0) = l(v_0)$. From 0-neighbourhood $(v_0)$ of $\mathcal{G}^\epsilon(D)$ , we obtain three 1-neighbourhoods: one $\forall r$-neighbourhood $(v_3)$ (since $V^\forall(v_0) = \{v_3\}$, $V^\epsilon(v_0) = \emptyset$) and two $r$-neighbourhoods $(v_1, v_3)$ and $(v_2, v_3)$ (since $(v_1 \epsilon v_3) \in E^\epsilon$, $p(v_1), p(v_3) \in \{v_0\}$ and $(v_2 \epsilon v_3) \in E^\epsilon$, $p(v_2), p(v_3) \in \{v_0\}$). Thus, we obtain two nodes $w_1, w_2 \in V'$ which are $r$-successors of $w_0$, and a node $w_3 \in V'$ which is $\forall r$-successor of $w_0$ where

$l'(w_1) = \mathsf{label}(v_1, v_3) = \{\{A\} \cup \{C\}\} = \{A, C\}$
$l'(w_2) = \mathsf{label}(v_2, v_3) = \{\{B\} \cup \{C\}\} = \{B, C\}$ and
$l'(w_3) = \mathsf{label}(v_3) = \{C\}$.

From 1-neighbourhood $(v_1, v_3)$, we obtain two 2-neighbourhoods: a $\forall r$-neighbourhood $(v_5)$ (since $V^\forall(v_1, v_3) = \{v_5\}$, $V^\epsilon(v_1, v_3) = \emptyset$) and $(v_4, v_5)$ (since $(v_4 \epsilon v_5) \in E^\epsilon$, $p(v_4), p(v_5) \in \{v_1, v_3\}$). Thus, we obtain two successors: $w_5$ is the $\forall r$-successor of $w_1$ where $l'(w_5) = \mathsf{label}(v_5) = \{C\}$ and $w_4$ is a $r$-successor of $w_1$ where $l'(w_4) = \mathsf{label}(v_4, v_5) = \{A, C\}$. Similarly, from 1-neighbourhood $(v_2, v_3)$, we obtain $\forall r$-neighbourhood $(v_6)$ and a $r$-neighbourhood $(v_4, v_6)$. Thus, we obtain two successors: $w_7$ is the $\forall r$-successor of $w_2$ where $l'(w_7) = \mathsf{label}(v_6) = \{B\}$ and $w_6$ is a $r$-successor of $w_2$ where $l'(w_6) = \mathsf{label}(v_4, v_6) = \{A, B\}$. Finally, from 1-neighbourhood $(v_3)$, we obtain $r$-neighbourhood

**Algorithm 1.** $\mathbf{B}(\mathcal{G}^\epsilon)$

**Require:** Graph $\mathcal{G}^\epsilon = (V, E \cup E^\epsilon, l)$ where $v^0$ is its root.

**Ensure:** Description tree $\mathbf{B}(\mathcal{G}^\epsilon) = (V', E', w^0, l')$

$V' := \emptyset$ $E' := \emptyset$.
Function $\phi$ from the set of subsets of $V$ into $V'$.
At level 0, for the unique 0-neighbourhood of $n^0$ of $\mathcal{G}^\epsilon$, we set $l'(w^0) := l(v^0)$ and $\phi(n^0) := w^0$.
**for all** $(k-1)$-neighbourhood $n^{k-1} \in N^{k-1}$ of $\mathcal{G}^\epsilon$ where $w^{k-1} = \phi(n^{k-1})$ **do**
  **if** there exists at least one node $v^{k-1} \in n^{k-1}$ such that $(v^{k-1} \forall r v^k) \in E$ **then**
    Let $n^k \in N(n^{k-1})$ be the $\forall r$-neighbourhood of $n^{k-1}$
    A node $w^k$ is created and $\phi(n^k) := w^k$
    $V' := V' \cup \{w^k\}$ and $E' := E' \cup \{(w^{k-1} \forall r w^k)\}$
    $l'(w^k) := \mathsf{label}(n^k)$
  **end if**
  **for all** $r$-neighbourhood $n^k \in N(n^{k-1})$ **do**
    A node $w^k$ is created and $\phi(n^k) := w^k$
    $V' := V' \cup \{w^k\}$ and $E' := E' \cup \{(w^{k-1} r w^k)\}$
    $l'(w^k) := \mathsf{label}(n^k)$
  **end for**
**end for**

$(v_4)$. Thus, we obtain a successor $w_8$ that is the $r$-successor of $w_3$ where $l'(w_8) = \mathsf{label}(v_4) = \{A\}$.

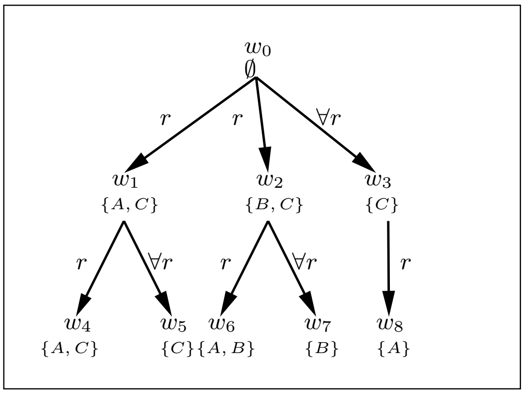


Figure 2. Description tree $\mathbf{B}(\mathcal{G}^\epsilon(D))$

As mentioned in Section 2, rules 5, 6, 7 (in Definition 2) need to be applied after applying rules 1, 2. This means that the normalization (by rules 1,

2, 5, 6, 7) makes explicit bottom-concept $\bot$ in the labels of nodes in the description tree. More precisely, rule 5 makes occurring bottom-concept $\bot$ in labels that contain $P$ and $\neg P$ for some $P \in N_C$. Rules 6, 7 propagate bottom-concept $\bot$ from nodes whose label contains bottom-concept $\bot$ to their $r$-ancestors (a node $w$ is called $r$-ancestor of a node $v$ if the path from $w$ to $v$ includes *only* $r$-edges).
From Algorithm 1, a neighbourhood in $\epsilon$-tree $\mathcal{G}^\epsilon$ corresponds to a node of description tree $\mathbf{B}(\mathcal{G}^\epsilon)$ i.e Algorithm 1 builds a mapping $\phi$ (cf. Algorithm 1) from the set of neighbourhoods to the nodes of the description tree. In particular, a path from a node $v^l$ to a node $v^k$: $(v^l r v^{l+1})$,...,$(v^{k-1} r v^k)$ $(l < k)$ in description tree $\mathbf{B}(\mathcal{G}^\epsilon)$ corresponds to the following neighbourhoods $n^l$, $n^{l+1}$,...,$n^k$ in $\mathcal{G}^\epsilon$ such that $n^{l+1} \in N(n^l)$, ..., $n^k \in N(n^{k-1})$, $\phi(n^l) = v^l$,...,$\phi(n^k) = v^k$ and $n^{m+1}$ is a $r$-neighbourhood of $n^m$ for all $l \le m < k$.
On the other hand, the application of rules 5, 6, 7 implies that the label of a node $v^l$ in the description tree (that corresponds to a weak normal concept description *after* applying rules 1 and 2) contains bottom-concept $\bot$ iff either the label of $v^l$ contains explicitly $\bot$ (or $P$ , $\neg P$ for some $P \in N_C$) or there exists a path composed of $r$-edges from $v^k$ to a node $v^l$ $(k > l)$ such that the label of $v^k$ contains explicitly $\bot$ (or $P$ , $\neg P$ for some $P \in N_C$).
From these remarks, we conclude that a neighbourhood $n^l$ in $\epsilon$-tree $\mathcal{G}^\epsilon$ corresponding to a node containing (explicitly and implicitly) bottom-concept $\bot$ in the tree $\mathbf{B}(\mathcal{G}^\epsilon)$ iff either the label of $n^l$ ($\mathsf{label}(n^l)$) contains explicitly $\bot$ (or $P$ , $\neg P$ for some $P \in N_C$) or there exist the following neighbourhoods $n^l$, $n^{l+1}$,...,$n^k$ in $\mathcal{G}^\epsilon$ such that $n^{l+1} \in N(n^l)$, ..., $n^k \in N(n^{k-1})$; $\phi(n^l) = v^l$,...,$\phi(n^k) = v^k$; $n^{m+1}$ is a $r$-neighbourhood of $n^m$ for all $l \le m < k$ and the label of $n^k$ contains explicitly $\bot$ (or $P$ , $\neg P$ for some $P \in N_C$). We say that such neighbourhoods $(n^l)$ in $\mathcal{G}^\epsilon$ contain a clash.
To sum up, we need only three nodes at the maximum to characterize each neighbourhood $n^k$ containing a clash in an $\epsilon$-tree $\mathcal{G}^\epsilon = (V, E \cup E^\epsilon, l)$. In fact,

1. If there is a node $v^k \in n^k$ such that $l(v^k) = \{\bot\}$ then $n^k$ together with all neighbourhoods containing $v^k$ contains a clash.
2. If there are two nodes $v_i^k, v_j^k \in n^k$ such that $P$, $\neg P \in l(v_i^k) \cup l(v_j^k)$ for some $P \in N_C$ then $n^k$ together with all neighbourhoods containing *both* $v_i^k, v_j^k$ contains a clash.
3. According to the remarks above, a neighbourhoods $n^l$ $(l < k)$ contains a clash if there exist $r$-neighbourhoods $n^{l+1}$,...,$n^k$ such that $n^{l+1} \in N(n^l)$, ..., $n^k \in N(n^{k-1})$ and $n^k$ contains a clash as described in 1. and 2. This means that the clash in $n^k$ is propagated along $r$-successors $v^l, ..., v^k$ in the neighbourhoods $n^l$,...,$n^k$. Thus, we need at the maximum two nodes for the propagated clash in $n^k$ and a node for the $r$-successor $v^l$ in $n^l$ to characterize the clash in the neighbourhood $n^l$.

These cases that are formalized in the following definition correspond, respectively, to 1-clash, 2-clash and 3-clash. We recall that predecessor function $p(v^k)$ defined as the $r$- predecessor or $\forall r$- predecessor of $v^k$ in $\epsilon$-trees, allows us to access to an ancestor $v^l$ of $v^k$ i.e $v^l = p^{k-l}(v^k)$.

**Definition 7** (clash) *Let $\mathcal{G}^\epsilon(C) = (V, E \cup E^\epsilon, l)$ be an $\epsilon$-tree. For all $v_i^k \in V$ such that $l(v_i^k) = \{\bot\}$, we define a* 1-clash*, denoted $[v_i^k]$.*
*For each pair of $\forall r$-successors $v_i^k, v_j^k \in V$ such that there exist $(v_i^k \epsilon v_j^k) \in E^\epsilon$ and $P$, $\neg P \in l(v_i^k) \cup l(v_j^k)$ for some $P \in N_C$, we define a* 2-clash*, denoted $[v_i^k, v_j^k]$.*
*Additionally, let $v_i^k, v_j^k \in V$ such that there exist $(v_i^k \epsilon v_j^k) \in E^\epsilon$ and $P$, $\neg P \in l(v_i^k) \cup l(v_j^k)$ for some $P \in N_C$ (or $l(v_i^k) = \{\bot\}$ if $v_i^k = v_j^k$). We define* clashes *from paths which are composed alternatively of $r$-edges and $\epsilon$-edges as follows. Let $v_1^l$, $v_1^{l+1}$, $v_2^{l+1}$,..., $v_1^{k-1}$, $v_2^{k-1}$, $v_2^k \in V$ $(l < k)$ such that the following conditions are satisfied:*

1. *there exists such a path from $v_1^l$ to $v_2^k$ i.e $(v_1^l r v_2^{l+1})$, $(v_2^{l+1} \epsilon v_1^{l+1})$, $(v_1^{l+1} r v_2^{l+2})$,..., $(v_2^{k-1} \epsilon v_1^{k-1})$, $(v_1^{k-1} r v_2^k) \in E \cup E^\epsilon$;*
2. *there exist $(v_1^l \epsilon p^{k-l}(v_i^k))$, $(v_1^l \epsilon p^{k-l}(v_j^k)) \in E^\epsilon$ and $(v_2^m \epsilon p^{k-m}(v_i^k))$, $(v_2^m \epsilon p^{k-m}(v_j^k)) \in E^\epsilon$ for all $m \in \{l+1, k\}$;*
3. *there do not exist nodes $v^l, v^{l-1} \in V$ such that $(v^l \epsilon v_1^l)$, $(v^l \epsilon p^{k-l}(v_i^k))$, $(v^l \epsilon p^{k-l}(v_j^k)) \in E^\epsilon$ and $(v^{l-1} r v_1^l) \in E$.*

*For each $m \in \{l, ..., k-1\}$ such that $v_1^m$, $p^{k-m}(v_i^k)$, $p^{k-m}(v_j^k)$ are $\forall r$-successors, we define:*

- *a* 3-clash*, denoted $[v_1^m, p^{k-m}(v_i^k), p^{k-m}(v_j^k)]$, if $v_1^m$, $p^{k-m}(v_i^k)$, $p^{k-m}(v_j^k)$ are pairwise different;*

– *a* 2-clash*, denoted* $[v_1^m, p^{k-m}(v_i^k)]$ *(or* $[v_1^m, p^{k-m}(v_j^k)]$*) if* $v_1^m \neq p^{k-m}(v_i^k)$ *(or* $v_1^m \neq p^{k-m}(v_j^k)$*);*
– *a* 1-clash*, denoted* $[v_1^m]$*, if* $v_1^m = p^{k-m}(v_i^k) = p^{k-m}(v_j^k)$*.*

Paths as described in Definition 7 (clash) can contain $\epsilon$-cycles if $v_2^m = v_1^m$ for some $m \in \{l-1, ..., k-1\}$. Furthermore, each clash $[v_1^k, ...v_q^k]$ can be contained in one or many neighbourhoods. These neighbourhoods are reachable by the propagation of bottom-concept $\perp$ via $r$-edges.

**Remark 3** *According to Definition* 7 (clash) *all 1-clashes, 2-clashes and 3-clashes are formed from one node, two nodes or three nodes. Therefore, the number of all these clashes is bounded by* $K \times |V| + K \times |V|^2 + K \times |V|^3$ *where* $|V|$ *is the cardinality of the set of nodes* $V$ *and* $K$ *is a constant.*

We show that the complexity of the computation for determining all 1-clashes , 2-clashes and 3-clashes, is polynomial in the size of $\mathcal{G}^\epsilon(C)=(V, E \cup E^\epsilon, l)$.
For each $\epsilon$-edge $(v_i^k \epsilon v_j^k)$ such that $P, \neg P \in \{l(v_i^k) \cup l(v_j^k)\}$ for some $P \in N_C$ (or $\perp \in l(v_i^k)$ if $v_i^k = v_j^k$), we obtain at level $(k-1)$ at most $|E|$ $r$-successors $v_{2,h}^{k-1}$ such that $(v_{2,h}^{k-1} \epsilon v_{1,h}^{k-1})$, $(v_{2,h}^{k-1} \epsilon p(v_i^k))$ and $(v_{2,h}^{k-1} \epsilon p(v_j^k))$ where $v_{1,h}^{k-1}$ are the $r$-predecessors of nodes $v_{2,h}^k$ (i.e $(v_{1,h}^{k-1} r v_{2,h}^k)$) and $(v_{2,h}^k \epsilon v_i^k)$ and $(v_{2,h}^k \epsilon v_j^k) \in E^\epsilon$.

Assume that at level $m$ we have $q_m$ $r$-successors $v_{2,h}^m$ where $1 \leq h \leq q_m$, $q_m \leq |E|$ such that $(v_{2,h}^m \epsilon v_{1,h}^m)$, $(v_{2,h}^m \epsilon p^{k-m}(v_i^k))$ and $(v_{2,h}^m \epsilon p^{k-m}(v_j^k))$. At level $(m-1)$, for each pair of nodes: $r$-successor $v_{2,h'}^{m-1}$ and $r$-predecessor $v_{1,h}^{m-1} = p(v_{2,h}^m)$ we check whether there exist $\epsilon$-edges $(v_{2,h'}^{m-1} \epsilon v_{1,h}^{m-1})$, $(v_{2,h'}^{m-1} \epsilon p^{k-m+1}(v_i^k))$ and $(v_{2,h'}^{m-1} \epsilon p^{k-m+1}(v_j^k))$. We will obtain $q_{m-1}$ such $r$-successors $v_{2,h'}^{m-1}$ where $1 \leq h' \leq q_{m-1}$, $q_{m-1} \leq |E|$, which satisfy the condition above. It means that there may be $\epsilon$-edges which connect a node $v_{2,h'}^{m-1}$ to many nodes $v_{1,h}^{m-1}$, $1 \leq h \leq q_m$. This computation process is stopped at level $(m-1)$ if there does not exist any $r$-successor $v_2^{m-1}$ such that $(v_2^{m-1} \epsilon v_{1,h}^{m-1})$, $(v_2^{m-1} \epsilon p^{k-m+1}(v_i^k))$ and $(v_2^{m-1} \epsilon p^{k-m+1}(v_i^k))$ for all $v_{1,h}^{m-1}$, $1 \leq h \leq q_m$. Therefore, at each level $m-1 \leq l < k$ we need at most $|V| \times (q_{l+1}+2)$ of checks to determine clashes at level $l$. Hence, we need at most $|V| \times (q_{l+1}+2) \times |\mathcal{G}^\epsilon(C)|$ of checks to compute all clashes triggered by an $\epsilon$-edge (that implies bottom-concept $\perp$). Since the number of $\epsilon$-edges is bounded by $|V|^2$, thus the complexity of the computation for determining all clashes is polynomial in $|V|$.

**Example 3** *Let* $C := \exists r.(\exists r.\forall r.\exists r.\top \sqcap \forall r.\forall r.\forall r.P) \sqcap \forall r.\forall r.\forall r.\forall r.(\neg P)$*. We have two clashes on* $\epsilon$*-tree* $\mathcal{G}^\epsilon(C)$ *as illustrated in Figure 3. According to Definition 7, first, there is a 2-clash* $[v_{10}, v_{11}]$ *since* $v_{10}$ *and* $v_{11}$ *are* $\forall r$*-successors such that there exists an* $\epsilon$*-edge* $(v_{10} \epsilon v_{11})$ *and* $P, \neg P \in l(v_{10}) \cup l(v_{11})$*. Second, we have 3-clash* $[v_6, v_7, v_8]$ *since there exist* $\epsilon$*-edges* $(v_9 \epsilon v_{10})$*,* $(v_9 \epsilon v_{11})$*,* $(v_{10} \epsilon v_{11})$ *and* $r$*-edge* $(v_6 r v_9)$ *such that* $P, \neg P \in l(v_{10}) \cup l(v_{11})$ *and there exist* $\epsilon$*-edge* $(v_6 \epsilon p(v_{10}))$ *and* $(v_6 \epsilon p(v_{11}))$*.*

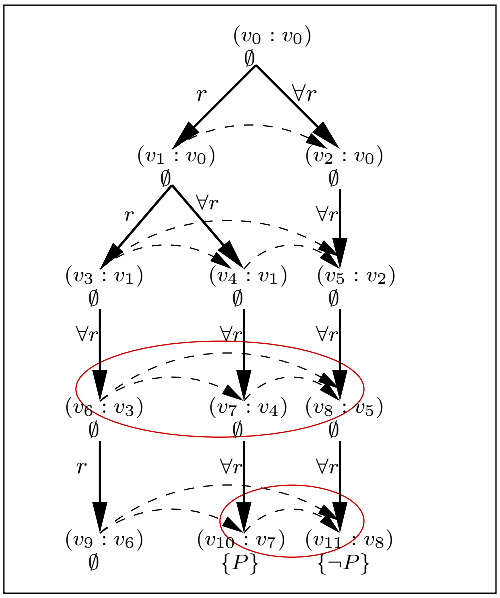


Figure 3. Clashes in $\epsilon$-tree $\mathcal{G}^\epsilon(C)$

The following lemma affirms that if a neighbourhood on an $\epsilon$-tree contains a clash then it must contain either 1-clash, 2-clash or 3-clash. This means that it is sufficient to use 1-clash, 2-clash, 3-clash for representing all clashes in $\epsilon$-trees.

**Lemma 1** *Let* $\mathcal{G}^\epsilon(C) = (V, E \cup E^\epsilon, l)$ *be an* $\epsilon$*-tree. A node* $v_1^l \in V$ *has a* q-clash $[v_1^l, .., v_q^l]$*,* $q \in \{1, 2, 3\}$ *such that* $\{v_1^l, .., v_q^l\} \subseteq n^l$ *where* $n^l$ *is a* $\forall r$*-neighbourhood iff one of the three following conditions is satisfied:*

1. *there exists* $v_i^l \in n^l$ *such that* $l(v_i^l) = \{\perp\}$*;*

2. *there exist $v_i^l, v_j^l \in n^l$ such that $v_i^l, v_j^l$ are $\forall r$-successors, $(v_i^l \epsilon v_j^l) \in E^\epsilon$ and there exists $P \in N_C$ such that $P$, $\neg P \in l(v_i^l) \cup l(v_j^l)$.*
3. 
   (a) *there exist nodes $v_i^k, v_j^k \in V$ $(l < k)$ such that $(v_i^k \epsilon v_j^k) \in E^\epsilon$. Moreover, if $v_i^k \neq v_j^k$ then there exists $P \in N_C$ such that $P$, $\neg P \in l(v_i^k) \cup l(v_j^k)$. If $v_i^k = v_j^k$ then $l(v_i^k) = \{\perp\}$; and*
   (b) *there exist neighbourhoods $n^l$, $n^{l+1} \in N(n^l)$, ... , $n^{k-1} \in N(n^{k-2})$, $n^k \in N(n^{k-1})$ and $r$-edges $(v_1^{m-1} r v_2^m) \in E$ such that $v_1^m, v_2^m \in n^m$ for all $l < m < k$, $(v_1^{k-1} r v_2^k) \in E$, $v_1^l \in n^l$ and $v_i^k, v_j^k, v_2^k \in n^k$.*

A proof of Lemma 1 can be found in Appendix.
Definition 7 (clash) allows us to detect all clashes in an $\epsilon$-tree. It does not show, however, how these clashes are stored in the $\epsilon$-tree. In Example 3, from the existence of 2-clash $[v_{10}, v_{11}]$ it is required that the label of all neighbourhoods containing nodes $v_{10}, v_{11}$ must contain a bottom-concept $\perp$. Thus, we need a node to store bottom-concept $\perp$ for this clash. However, it is not possible to use, for example, node $v_{11}$ for this purpose since $v_{11}$ belongs to neighbourhood $(v_{11})$ which does not include *both* nodes $v_{10}, v_{11}$. In addition, from 3-clash $[v_6, v_7, v_8]$ it is required that the label of all neighbourhoods containing nodes $v_6, v_7, v_8$ must contain a bottom-concept $\perp$. Therefore, we need node(s) to store bottom-concept $\perp$ for this clash. Similarly, we cannot store a bottom-concept $\perp$ to the label of node $v_8$.
This problem is resolved by exploiting the neighbourhood notion given Definition 6 (set $V^\epsilon$ is introduced to this definition for this purpose) and proposing a structure, called *normalization graph*. This structure, denoted $\mathcal{G}_C^\epsilon$, is extended from $\epsilon$-tree $\mathcal{G}^\epsilon(C)$ by adding some nodes and edges such that it can store bottom-concepts $\perp$ for clashes $[v_1^k, .., v_q^k]$ $(1 \leq q \leq 3)$ in preserving the other neighbourhoods in $\epsilon$-tree $\mathcal{G}^\epsilon(C)$. It is necessary to add new nodes in order to store bottom-concept $\perp$ for q-clash since, as explained above, nodes in $\epsilon$-trees can be shared by several neighbourhoods and a node belonging to a q-clash may belong to a neighbourhood which does not include this q-clash.
To sum up, the neighbourhood notion for the normalization graph extended from a $\epsilon$-tree has to be defined such that i) it preserves the neighbourhoods in the $\epsilon$-tree if these neighbourhoods do not contain any clash, and ii) it yields a new neighbourhood for each neighbourhood contains a clash in the $\epsilon$-tree . The new neighbourhoods, represented as sets $V^\epsilon$ in the Definition 6 (neighbourhood), are formed from the added nodes whose label contains bottom-concept $\perp$. The neighbourhood notion defined in this way allows not only for guaranteeing a correct transformation (by Algorithm 1) from normalization graphs into description trees but also extending naturally the product operation of description trees presented in [3] to the product operation of normalization graphs. More precisely, we consider the three following cases when constructing the normalization graph from an $\epsilon$-tree (two of them are illustrated in Figure 4):

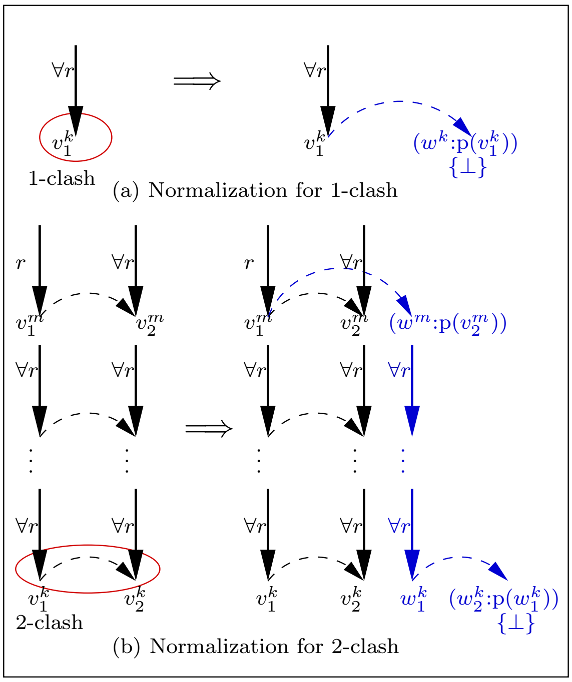


Figure 4. Normalization for clashes

1. Let $[v_1^k]$ be a 1-clash (Figure 4 (a)). According to Definition 7 (clash), $v_1^k$ has to be a $\forall r$-successor. In this case, a node $w^k$ is added and $l(w^k) = \{\perp\}$, $p(w^k) = p(v_1^k)$. Furthermore, an $\epsilon$-edge $(v_1^k \epsilon w^k)$ is also added to guarantee that, according to Definition 6 (neighbourhood), any $(k-1)$-neighbourhood $n^{k-1}$ in the normalization graph $\mathcal{G}_C^\epsilon$ such that $v_1^k \in V^\forall(n^{k-1})$ iff $w^k \in V^\epsilon(n^{k-1})$.

2. Let $[v_1^k, v_2^k]$ be a 2-clash (Figure 4 (b)). According to Definition 7 (clash), $v_1^k, v_2^k$ have to be $\forall r$-successors. Since $\mathcal{G}^\epsilon(C)$ is an $\epsilon$-tree obtained from $C$ in weak normal form, every node has at most a $\forall r$-successor. This implies that there exists a unique $r$-successor $v_i^m$ such that $m < k$ ($m$ is the highest level from the root), $v_i^m = p^{k-m}(v_1^k)$ or $v_i^m = p^{k-m}(v_2^k)$ ($v_i^m = v_1^m = p^{k-m}(v_1^k)$ in the figure). In this case, nodes $w^m$, ..., $w_1^k$, $w_2^k$ and edges connecting them together are added. Note that, $p(w^m)$ and $p(w_2^k)$ are set to $p(v_2^m)$ and $p(w_1^k)$, respectively. This guarantees that, according to Definition 6 (neighbourhood), $v_1^k, v_2^k \in V^\forall(n^{k-1})$ iff $w_2^k \in V^\epsilon(n^{k-1})$ for any $(k-1)$-neighbourhood $n^{k-1}$ in the normalization graph $\mathcal{G}_C^\epsilon$.
3. Let $[v_1^k, v_2^k, v_3^k]$ be a 3-clash. Similarly, $v_1^k, v_2^k, v_3^k$ have to be $\forall r$-successors. Since $\mathcal{G}^\epsilon(C)$ is an $\epsilon$-tree, there exists a unique $r$-successor $v_i^m$ such that $m < k$ ($m$ is the highest level from the root) and $p^{k-m}(v_i^k) = v_i^m$ where $v_i^k \in \{v_1^k, v_2^k, v_3^k\}$. In this case, nodes $w^m$, ..., $w_1^k$, $w_2^k$ and edges connecting them together are added. Note that, differently from the case for 2-clash, $p(w^m)$ and $p(w_2^k)$ are set respectively to $p(v_j^m)$ and $p(v_l^k)$ where $p^{k-m}(v_j^k) = v_j^m$, $v_j^k \in \{v_1^k, v_2^k, v_3^k\} \setminus \{v_i^k\}$ and $v_l^k \in \{v_1^k, v_2^k, v_3^k\} \setminus \{v_i^k, v_j^k\}$. This guarantees that, according to Definition 6 (neighbourhood), $v_1^k, v_2^k, v_3^k \in V^\forall(n^{k-1})$ iff $w_2^k \in V^\epsilon(n^{k-1})$ for any $(k-1)$-neighbourhood $n^{k-1}$ in the normalization graph $\mathcal{G}_C^\epsilon$ (cf. Figure 5).

Definition 8 (normalization graph) formalizes the procedure described above.
Before normalizing an $\epsilon$-tree $\mathcal{G}^\epsilon(C) = (V, E \cup E^\epsilon, l)$ by Definition 8 (normalization graph), $\mathcal{G}^\epsilon(C)$ can be simplified as follows:

– Let $[v_1^l, ..., v_a^l]$ and $[w_1^k, ..., w_b^k]$ be clashes in $\mathcal{G}^\epsilon(C)$ such that $l < k$ and $p^{k-l}(w_i^k) \in \{v_1^l, ..., v_a^l\}$ for all $w_i^k \in \{w_1^k, ..., w_b^k\}$. From the definition of neighbourhood, it holds that for each $k$-neighbourhood $n^k$ such that $\{w_1^k, ..., w_b^k\} \subseteq n^k$, there exist neighbourhoods $n^l$, ..., $n^{k-1}$ such that $n^l$, $n^{l+1} \in N(n^l)$, ... , $n^k \in N(n^{k-1})$ and $\{v_1^l, ..., v_b^l\} \subseteq n^l$.
From this claim, if $[v_1^k]$ be a 1-clash in $\mathcal{G}^\epsilon(C)$ then the subtree $\mathcal{G}^\epsilon(C)(v_1^k)$ can be deleted from $\mathcal{G}^\epsilon(C)$. This can be performed by deleting all nodes $v^h \in V$, $h > k$ such that $p^{h-k} = v_1^k$ and all edges such that one of two endpoints belongs to the set of deleted nodes.

**Definition 8** (*normalization graph*) *Let $\mathcal{G}^\epsilon(C) = (V, E \cup E^\epsilon, l)$ be an $\epsilon$-tree and $v^0$ is its root. The normalization graph of $C$, denoted $\mathcal{G}_C^\epsilon = (V', E' \cup E'^\epsilon, l')$, is obtained from $\mathcal{G}^\epsilon(C)$ by adding nodes and edges as follows. If the root $v^0$ has a* 1-clash *$[v^0]$ then $l'(v^0) := \{\perp\}$. For each $v_1^k \in V$, $v_1^k \notin v^0$ such that $v_1^k$ has a* q-clash *$[v_1^k, .., v_q^k]$ $(1 \leq q \leq 3)$ and there does not exist any* clash *$[w_1^k, .., w_p^k]$ such that $\{w_1^k, .., w_p^k\} \subseteq \{v_1^k, .., v_q^k\}$,*

1. *If $q = 1$ (i.e $[v_1^k, .., v_q^k]$=$[v_1^k]$) then we add to $\mathcal{G}^\epsilon(C)$ a node $w^k$ with an $\epsilon$-cycle and edge $(v_1^k \epsilon w^k)$ where $l'(w^k)$:=$\{\perp\}$ and $p(w^k) := p(v_1^k)$. Additionally, the node $v_1^k$ is relabeled with $\emptyset$.*
2. *Let $m < k$ be the highest level from the root $v^0$ such that there exists a $r$-successor $v_i^m = p^{k-m}(v_i^k)$ where $v_i^k \in \{v_1^k, .., v_q^k\}$. We add to $\mathcal{G}^\epsilon(C)$ nodes $w^m$,..., $w^{k-1}, w_1^k, w_2^k$ with $\epsilon$-cycles and edges $(v_i^m \epsilon w^m)$, $(w^m \forall r w^{m-1})$ ,..., $(w^{k-1} \forall r w_1^k)$, $(w_1^k \epsilon w_2^k)$,*
   (a) *If $q = 2$ (i.e $[v_1^k, .., v_q^k]$=$[v_1^k, v_2^k]$) then the labels and predecessors of the added nodes are initialized as follows: $l'(w_2^k)$:= $\{\perp\}$ and $l'(w^m)$= ... =$l'(w_1^k)$:=$\emptyset$; $p(w_1^k) = p(w_2^k) := w^{k-1}$, $p(w^{k-1}) := w^{k-2}$, ..., $p(w^m)$:=$p^{k-m+1}(v_j^k)$ where $v_j^k \in \{v_1^k, v_2^k\} \setminus \{v_i^k\}$;*
   (b) *If $q = 3$ (i.e $[v_1^k, .., v_q^k] = [v_1^k, v_2^k, v_3^k]$) then the labels and predecessors of the added nodes are initialized as follows: $l'(w_2^k)$:= $\{\perp\}$ and $l'(w^m)$= ... =$l'(w_1^k)$:=$\emptyset$; $p(w_1^k)$ := $w^{k-1}$, $p(w^{k-1})$:= $w^{k-2}$,..., $p(w^m)$:= $p^{k-m+1}(v_j^k)$ where $v_j^k \in \{v_1^k, v_2^k, v_3^k\} \setminus \{v_i^k\}$; and $p(w_2^k) := p(v_l^k)$ where $v_l^k \in \{v_1^k, v_2^k, v_3^k\} \setminus \{v_i^k, v_j^k\}$.*

The root of normalization graph $\mathcal{G}_C^\epsilon$ is the root of the $\epsilon$-tree $\mathcal{G}^\epsilon(C)$. The level of a node is defined as the number of ordinary edges of a path from the root to that node since the number of ordinary edges of all paths from the root to a node is constant. In normalization graphs the predecessor $p(v^k)$ of a node $v^k$ may not be a $r$-successor and $\forall r$-successor.

**Remark 4** *According to Remark 3, the number of* clashes *in a $\epsilon$-tree $\mathcal{G}^\epsilon(C)$ is bounded $K \times |V| +$*

*$K \times |V|^2 + K \times |V|^3$ where $|V|$ is the cardinality of $V$ and $K$ is a constant. Furthermore, the number of nodes and edges are added by Definition 8 (normalization graph) for each* clash *is bounded by $|\mathcal{G}^\epsilon(C)|$. Thus, the size of normalization graph $\mathcal{G}^\epsilon_C$ increases polynomially in the size of $C$.*

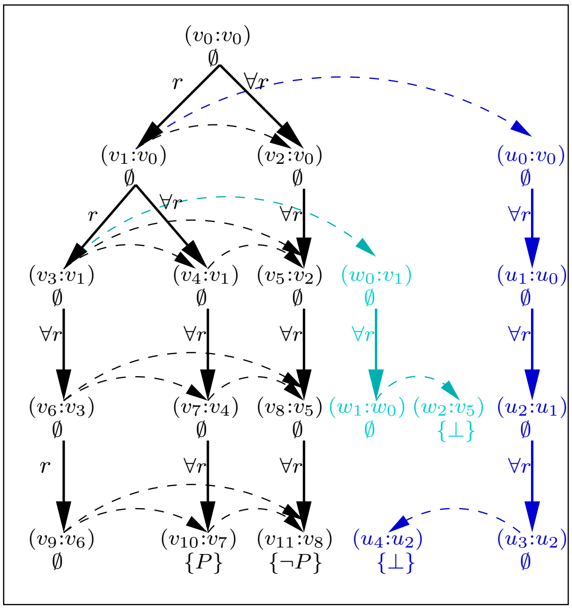


Figure 5. Normalization graph $\mathcal{G}^\epsilon_C$

Figure 5 illustrates the normalization graph corresponding to the $\epsilon$-tree in Example 3 (Figure 3). In this figure, the nodes $u_0$, $u_1$, $u_2$, $u_3$, $u_4$, $\forall r$-edges $(u_0 \forall r u_1)$,..., $(u_3 \forall r u_4)$ and $\epsilon$-edges $(v_1 \epsilon u_0)$, $(u_3 \epsilon u_4)$ are added for 2-clash $[v_{10}, v_{11}]$. Moreover, $p(u_0) = p(v_2) = v_0$ and $p(u_4) = p(u_3) = u_2$. Next, the nodes $w_0$, $w_1$, $w_2$ , $\forall r$-edge $(w_0 \forall r w_1)$ and $\epsilon$-edges $(v_3 \epsilon w_0)$, $(w_1 \epsilon w_2)$ are added for 3-clash $[v_6, v_7, v_8]$. Moreover, $p(w_0) = p(v_4) = v_1$ and $p(w_2) = p(v_8) = v_5$.

As a result, we obtain that i) $v_{10}, v_{11} \in V^\forall(n^3)$ iff $u_4 \in V^\epsilon(n^3)$ for some 3-neighbourhood $n^3$ in the normalization graph $\mathcal{G}^\epsilon_C$ (Figure 5), ii) $v_7, v_8 \in V^\forall(n^1)$ iff $w_2 \in V^\epsilon(n^1)$ for some 1-neighbourhood $n^1$ in the normalization graph $\mathcal{G}^\epsilon_C$.

For example, we show i). Assume that $v_{10}, v_{11} \in V^\forall(n^3)$ for some 3-neighbourhood $n^3$. Since $p^3(v_{10}) = v_1$, $p^3(v_{11}) = v_2$ there exist neighbourhoods $n^1$, $n^2$ in $\mathcal{G}^\epsilon_C$ such that $v_1, v_2 \in n^1$ and $n_2 \in N(n_1)$, $n_3 \in N(n_2)$. This implies that $u_0 \in n^1$. Note that, by the construction of $\mathcal{G}^\epsilon_C$ it holds that $u_0 \in n^1$ iff $v_1, v_2 \in n^1$ (*). Hence, we obtain that $v_{10}, v_{11}, u_3 \in V^\forall(n^3)$. Since $p(u_4) \in n^3$, $(u_3 \epsilon u_4)$ and $u_4 \notin V^\forall(n^3)$, according to Definition 6 (neighbourhood), $u_4 \in V^\epsilon(n^3)$. Conversely, assume that $u_4 \in V^\epsilon(n^3)$. We have that $u_2 \in n^3$ since $p(u_4) \in n^3$, $(u_3 \epsilon u_4) \in E^\epsilon$ and $(u_2 \forall r u_3) \in E$. Let $n^1$ be a 1-neighbourhood in $\mathcal{G}^\epsilon_C$ such that $u_0 \in n^1$. Since $p^2(u_2) = u_0$ hence there exists a neighbourhood $n^2$ such that $n_2 \in N(n_1)$, $n_3 \in N(n_2)$. Moreover, (*) yields that if $u_0 \in n^1$ then $v_1, v_2 \in n^1$ (**). From (**) and $p^3(v_{10}) = v_1$, $p^3(v_{11}) = v_2$, we obtain that $v_{10}, v_{11} \in V^\forall(n^3)$.

To sum up, the normalization graph $\mathcal{G}^\epsilon_C$ preserves all neighbourhoods in the $\epsilon$-tree $\mathcal{G}^\epsilon(C)$ and yields new neighbourhoods, represented by $V^\epsilon$, which correspond to neighbourhoods containing clashes in $\mathcal{G}^\epsilon(C)$. The following lemma generalizes this important property of normalization graphs.

**Lemma 2** *Let $\mathcal{G}^\epsilon(C) = (V, E \cup E^\epsilon, l)$ be an $\epsilon$-tree and $\mathcal{G}^\epsilon_C = (V', E' \cup E'^\epsilon, l')$ be its normalization graph. Let $n'^k$ be a $r$-neighbourhood ($\forall r$-neighbourhood) in $\mathcal{G}^\epsilon_C$.*

1. label$(n'^k) \neq \{\perp\}$ *iff*
   (a) *$n'^k$ is the 0-neighbourhood and $v^0$ does not have any* clash, *or*
   (b) *there exists a $r$-neighbourhood ($\forall r$-neighbourhood) $n^k$ in $\mathcal{G}^\epsilon(C)$ such that $n^k = n'^k \cap V$ and* label$(n^k)$ = label$(n'^k)$.
2. label$(n'^k) = \{\perp\}$ *iff*
   (a) *$n'^k$ is the 0-neighbourhood and $v^0$ has a* 1-clash $[v^0]$, *or*
   (b) *there exists a* q-clash $[v^k_1, .., v^k_q]$ *such that $\{v^k_1, .., v^k_q\} \subseteq n^k$, $n^k = V^\forall(n'^{k-1}) \cap V$ and $n'^k \in N(n'^{k-1})$ where $n^k$ is a $\forall r$-neighbourhood in $\mathcal{G}^\epsilon(C)$.*

A proof of Lemma 2 can be found in Appendix. Algorithm 1 can transform $\epsilon$-trees $\mathcal{G}^\epsilon(C)$ and normalization graphs $\mathcal{G}^\epsilon_C(C)$ into description trees since the neighbourhood, level, predecessor notions are well defined for both graphs. In general, the description tree $\mathbf{B}(\mathcal{G}^\epsilon_C)$ is different from $\mathbf{B}(\mathcal{G}^\epsilon(C))$. However, $\mathbf{B}(\mathcal{G}^\epsilon_C)$ can be obtained from $\mathbf{B}(\mathcal{G}^\epsilon(C))$ by normalization rules which are defined for description trees. These rules must correspond to rules 5, 6, 7 (Definition 2) defined for concept descriptions.

**Lemma 3** *Let $C$ be an $\mathcal{ALE}$-concept description in the weak normal form and $\mathcal{G}^\epsilon(C)$ be its description* *tree. There exists an isomorphism between* $\mathbf{B}(\mathcal{G}^\epsilon_C)$ *and the description tree* $\mathcal{H}$ *obtained from* $\mathbf{B}(\mathcal{G}^\epsilon(C))$ $= (V_3, E_3, z^0, l_3)$ *by exhaustively applying the following rules:*

1. $P, \neg P \in l_3(z)$, $P \in N_C \rightarrow$
   $l_3(z) := \{\bot\}$ *(rule 5g)*
2. $(zrz') \in E_3$, $\mathbf{B}(\mathcal{G}^\epsilon(C))(z') = \mathcal{G}(\bot) \rightarrow$
   $\mathbf{B}(\mathcal{G}^\epsilon(C))(z) := \mathcal{G}(\bot)$ *(rule 6g)*
3. $\bot \in l_3(z) \rightarrow$
   $\mathbf{B}(\mathcal{G}^\epsilon(C))(z) := \mathcal{G}(\bot)$ *(rule 7g)*

A proof of Lemma 3 can be found in Appendix.
The following proposition is an important result of this section. It establishes the equivalence between the normalization by the rules in Definition 2 for concept descriptions, and the normalization by Definitions 5 and 8 for description trees.

**Proposition 1** *Let $C$ be an $\mathcal{ALE}$-concept description. Let $\mathcal{G}_C$ and $\mathcal{G}^\epsilon_C$ be its description tree and normalization graph, respectively. There exists an isomorphism between $\mathbf{B}(\mathcal{G}^\epsilon_C)$ and $\mathcal{G}_C$.*

A proof of Proposition 1 can be found in Appendix.

**Remark 5** *Proposition 1 and Example* 1 *yield that the size of $\mathbf{B}(\mathcal{G}^\epsilon_C)$ may be exponential in the size of $\mathcal{G}^\epsilon_C$.*

In order to terminate this section, we exploit the obtained results to propose a polynomial algorithm in space for deciding subsumption between two $\mathcal{ALE}$-concept descriptions. This algorithm manipulates directly normalization graphs $\mathcal{G}^\epsilon$ and $\mathcal{H}^\epsilon$ to check the existence of a homomorphism between description trees $\mathbf{B}(\mathcal{H}^\epsilon)$ and $\mathbf{B}(\mathcal{G}^\epsilon)$.
Note that the algorithm described in [3] for checking the existence of a homomorphism between two $\mathcal{ALE}$-description trees obtained from *normalized* concept descriptions cannot be used for this aim since it requires that all nodes of description trees is explicitly represented *i.e.* it requires an exponential space.
The underlying idea of Algorithm 2 is that checking the existence of a homomorphism from $\mathbf{B}(\mathcal{H}^\epsilon)$ to $\mathbf{B}(\mathcal{G}^\epsilon)$ can be performed without transforming completely $\mathcal{H}^\epsilon$ and $\mathcal{G}^\epsilon$ into $\mathbf{B}(\mathcal{H}^\epsilon)$ and $\mathbf{B}(\mathcal{G}^\epsilon)$. By fixing on each neighbourhood of $\mathcal{H}^\epsilon$ from the highest level to the root, this process can be carried out by checking the existence of a mapping between *neighbourhood paths* from the root to neighbourhoods at the highest level in $\mathcal{H}^\epsilon$ and $\mathcal{G}^\epsilon$. At each

**Algorithm 2.** $\mathsf{check}(\mathcal{H}^\epsilon(n^k),\ \mathcal{G}^\epsilon(m^k))$

**Require:** $n^k, m^k$ are $k$-neighbourhoods, respectively, in normalization graphs $\mathcal{H}^\epsilon$ and $\mathcal{G}^\epsilon$.
**Ensure:** Answer "**true**" if there exists a homomorphism from $\mathcal{H}^\epsilon$ to $\mathcal{G}^\epsilon$. Otherwise, answer "**false**".
  **if** $\mathsf{label}(m^k) = \{\bot\}$ **then**
    **return true**;
  **end if**
  **if** $\mathsf{label}(n^k) \not\subseteq \mathsf{label}(m^k)$ **then**
    **return false**;
  **end if**
  Let $n^{k+1}_1, ..., n^{k+1}_p$ be $(k+1)$-neighbourhoods generated from $n^k$;
  Let $m^{k+1}_1, ..., m^{k+1}_q$ be $(k+1)$-neighbourhoods generated from $m^k$;
  **for** $1 \le i \le p$ **do**
    found := **false**;
    **for** $1 \le j \le q$ **do**
      **if** $n^{k+1}_i$, $m^{k+1}_j$ are $\forall r$-neighbourhoods **and** $\mathsf{check}(\mathcal{H}^\epsilon(n^{k+1}_i),\ \mathcal{G}^\epsilon(m^{k+1}_j))$ **then**
        found := **true**;
      **end if**
      **if** $n^{k+1}_i$, $m^{k+1}_j$ are $r$-neighbourhoods **and** $\mathsf{check}(\mathcal{H}^\epsilon(n^{k+1}_i),\ \mathcal{G}^\epsilon(m^{k+1}_j))$ **then**
        found := **true**;
      **end if**
    **end for**
    **if** found = **false** **then**
      **return false**;
    **end if**
  **end for**
  **return true**;

checking step, the algorithm needs memory pieces to unfold neighbourhood paths in $\mathcal{H}^\epsilon$ and $\mathcal{G}^\epsilon$.

Let $m^0$, $n^0$ be 0-neighbourhoods of, respectively, normalization graphs $\mathcal{H}^\epsilon$, $\mathcal{G}^\epsilon$. If function **check**$(\mathcal{H}^\epsilon(n^0),\ \mathcal{G}^\epsilon(m^0))$ returns "**true**", there exists a homomorphism from $\mathbf{B}(\mathcal{H}^\epsilon)$ to $\mathbf{B}(\mathcal{G}^\epsilon)$. Otherwise, there does not exist any homomorphism from $\mathbf{B}(\mathcal{H}^\epsilon)$ to $\mathbf{B}(\mathcal{G}^\epsilon)$.

**Completeness of the algorithm**
Assume that there exists a homomosphism $\varphi$ from $\mathbf{B}(\mathcal{H}^\epsilon)$ to $\mathbf{B}(\mathcal{G}^\epsilon)$. We have to show that **check**$(\mathcal{H}^\epsilon(n^0),\ \mathcal{G}^\epsilon(m^0))$ returns "**true**".
Let $\{w'_0, ..., w'_n\}$ be a post-order sequence of nodes of $\mathbf{B}(\mathcal{H}^\epsilon)$ (note that node $w'_n$ corresponds to the

root of $\mathcal{H}^\epsilon$). This sequence corresponds to a sequence of neighbourhoods of $\mathcal{H}^\epsilon$. If there is not any confusion, we can read $w'_n$ for a neighbourhood on $\mathcal{H}^\epsilon$. We will prove the claim by induction on $m$ where $0 \leq i \leq n$.

- Step $i = 0$. Let $v = \varphi(w'_0)$. Since $\varphi$ is a homomorphism, it is that $l(w'_0) \subseteq l(\varphi(w'_0))$ or $l(\varphi(w'_0)) = \{\perp\}$ ($l(w'_i) = \mathsf{label}(n_i^k)$ where $n_i^k$ is the neighbourhood corresponding to $w'_i$). Furthermore, since $p, q$ are equal to zero in the algorithm, no iteration is performed. Thus, $\mathbf{check}(\mathcal{H}^\epsilon(w'_0), \mathcal{G}^\epsilon(\varphi(w'_0)))$ returns "**true**".
- Induction step $(i-1) \rightarrow i$. By induction hypothesis, $\mathbf{check}(\mathcal{H}^\epsilon(w'_j), \mathcal{G}^\epsilon(\varphi(w'_j)))$ returns "**true**" for all $0 \leq j < i$. Let $v = \varphi(w'_i)$. Since $\varphi$ is a homomorphism, it is $l(v) = \{\perp\}$ or $l(w'_i) \subseteq l(v)$ . Let $w'_{i_1}, ..., w'_{i_p}$ be the neighbourhoods generated from the neighbourhood $w'_i$ and the edges $(w'_i r_1 w'_{i_1})$,...,$(w'_i r_p w'_{i_p})$. Since $\{w'_0, ..., w'_n\}$ is a post-order sequence, hence $i_1, ...i_p \in \{0, ..., i-1\}$. By induction hypothesis, $\mathbf{check}(\mathcal{H}^\epsilon(w'_{i_l}), \mathcal{G}^\epsilon(\varphi(w'_{i_l})))$ returns "**true**" for all $1 \leq l \leq p$. Since $\varphi$ is a homomorphism and $w'_{i_1}, ..., w'_{i_p}$ are the neighbourhoods generated from the neighbourhood $w'_i$, hence $\varphi(w'_{i_j})$ have to be the neighbourhoods generated from the neighbourhood $v$ and the edges $(v r_l \varphi(w'_{i_j}))$ for all $1 \leq l \leq p$. This implies that $\mathbf{check}(\mathcal{H}^\epsilon(w'_i), \mathcal{G}^\epsilon(v))$ returns "**true**" since the iteration with index $j$ (second iteration) in the algorithm does not return "**false**" for all $1 \leq j \leq q$.

**Soundness of the algorithm**
Assume that $\mathbf{check}(\mathcal{H}^\epsilon(w_0), \mathcal{G}^\epsilon(v_0))$ returns "**true**". We have to show that there exists a homomorphism $\varphi$ from $\mathbf{B}(\mathcal{H}^\epsilon)$ to $\mathbf{B}(\mathcal{G}^\epsilon)$.
Since $\mathbf{check}(\mathcal{H}^\epsilon(w_0), \mathcal{G}^\epsilon(v_0))$ returns "**true**", it is that $l(w_0) \subseteq l(v_0)$ or $l(v_0) = \{\perp\}$. We start with $\varphi(w_0) := v_0$. Let $w^1, ..., w^p$ be neighbourhoods generated from the neighbourhood $w_0$ and $v^1, ..., v^q$ be neighbourhoods generated from the neighbourhood $v_0$. Since $\mathbf{check}(\mathcal{H}^\epsilon(w_0), \mathcal{G}^\epsilon(v_0))$ returns "**true**", according to the algorithm there exist $v^{i_l} \in \{v^1, ..., v^q\}$ such that $l(v^{i_l}) = \{\perp\}$ or $l(w^l) \subseteq l(v^{i_l})$, the edges $(w_0 r_l w^l))$, $(v_0 r_l v^{i_l}))$ and $\mathbf{check}(\mathcal{H}^\epsilon(w^l), \mathcal{G}^\epsilon(v^{i_l}))$ returns "**true**" for all $1 \leq l \leq p$. We define $\varphi(w^l) := v^{i_l}$ for all $1 \leq l \leq p$. This process goes on to leaves of $\mathbf{B}(\mathcal{H}^\epsilon)$ from the fact that $\mathbf{check}(\mathcal{H}^\epsilon(w^l), \mathcal{G}^\epsilon(v^{i_l}))$ returns "**true**" for all $1 \leq l \leq p$. This implies that homomorphism $\varphi$ from $\mathbf{B}(\mathcal{H}^\epsilon)$ to $\mathbf{B}(\mathcal{G}^\epsilon)$ is defined.

**Proposition 2** *Let $C$ and $D$ be $\mathcal{ALE}$-concept descriptions, and let $\mathcal{G}^\epsilon_C$ and $\mathcal{G}^\epsilon_D$ be their normalization graphs. Algorithm 2 applied to $\mathcal{G}^\epsilon_C$ and $\mathcal{G}^\epsilon_D$ can decide subsumption between $C$ and $D$ in polynomial space and exponential time.*

A proof of Proposition 2 can be found in Appendix.

## 4. Product of normalization graphs

This section introduces the product operation of normalization graphs, which is extended from the product operation of description trees (as defined in [3]). In this extension, $\epsilon$-edges including $\epsilon$-cycles will be treated as ordinary edges. In particular, for an $\epsilon$-cycle $(v\epsilon v)$, we say also that $v$ is an $\epsilon$-successor of itself.
Additionally, we need the notion of induced subgraph to treat nodes whose label is equal to $\{\perp\}$. An *induced subgraph* $\mathcal{G}^\epsilon(v_1^k, ..., v_m^k)$ of graph $\mathcal{G}^\epsilon$ where $v_1^k, ..., v_m^k$ are nodes at level $k$ of $\mathcal{G}^\epsilon$, consists of the set of nodes $v_1^k, ..., v_m^k$ and their descendants in $\mathcal{G}^\epsilon$ together with all edges whose endpoints are both in this set of nodes. More precisely, let $\mathcal{G}^\epsilon = (V_G, E_G \cup E^\epsilon_G, l_G)$ and $v_1^k, ..., v_m^k \in V_G$. We define an induced subgraph $\mathcal{G}^\epsilon(v_1^k, ..., v_m^k) = (V_{G_k}, E_{G_k} \cup E^\epsilon_{G_k}, l_{G_k})$ where
$V_{G_k} := \{v^l \in V_G |\ p^{l-k}(v^l) \in \{v_1^k, ..., v_m^k\}, l \geq k\}$,
$E_{G_k} := \{(v^l e v^{l+1}) \in V_G |\ v^l, v^{l+1} \in V_{G_k}\}$,
$E^\epsilon_{G_k} := \{(v^l \epsilon v'^l) \in E^\epsilon_G |\ v^l, v'^l \in V_{G_k}\}$, and
$l_{G_k}(v) := l_G(v)$ for all $v \in V_{G_k}$.
Note that, from the definition of predecessor function $p$ for normalization graphs $\mathcal{G}^\epsilon$, a node $v^l \in V_G$ is not a $r$-successor and $\forall r$-successor but $v^l$ may belong to an induced subgraph $\mathcal{G}^\epsilon(v_1^k, ..., v_m^k)$ if $p^{l-k}(v^l) \in \{v_1^k, ..., v_m^k\}$.
In order to transform a subgraph $\mathcal{G}^\epsilon(v_1^k, ..., v_m^k)$ into a description tree, we can apply Algorithm 1 to the graph obtained from $\mathcal{G}^\epsilon(v_1^k, ..., v_m^k)$ by replacing the nodes $v_1^k$,..., $v_m^k$ with a unique node $v_0$ which is considered as the root of the subgraph. The label of $v_0$ is set to $\mathsf{label}(v_1^k, ..., v_m^k)$, the outgoing edges of $v_1^k$,..., $v_m^k$ become those of $v_0$, and the $\epsilon$-edges between nodes $v_1^k$,..., $v_m^k$ become the $\epsilon$-cycle of $v_0$. In particular, if $\{v_1^k, ..., v_m^k\}$ is a $k$-neighbourhood in $\mathcal{G}^\epsilon$ then $\mathcal{G}^\epsilon(v_1^k, ..., v_m^k)$ contains all $l$-neighbourhoods generated from $\{v_1^k, ..., v_m^k\}$ where $l \geq k$. Algorithm 1 yields that the nodes and edges of the tree $\mathbf{B}(\mathcal{G}^\epsilon(v_1^k, ..., v_m^k))$ can be obtained from these neighbourhoods.

**Definition 9** *Let* $\mathcal{G}^\epsilon = (V_G, E_G \cup E^\epsilon_G, l_G)$*,* $\mathcal{H}^\epsilon = (V_H, E_H \cup E^\epsilon_H, l_H)$ *be two normalization graphs where* $v^0$ *and* $w^0$ *are the roots respectively of* $\mathcal{G}^\epsilon$ *and* $\mathcal{H}^\epsilon$*. If* $l_G(v^0)=\{\bot\}$ $(l_H(w^0)=\{\bot\})$ *then we define* $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$ *as a graph obtained from* $\mathcal{H}^\epsilon$ $(\mathcal{G}^\epsilon)$ *by replacing each node* $w \in V_H$ $(v \in V_G)$ *with* $(v^0, w)$ $((v, w^0))$ *where* $(v^0, w)$ $((v, w^0))$ *is labeled with* $l_G(v^0) \cap l_H(w)$ $(l_G(v) \cap l_H(w^0))$*. Otherwise, the node* $(v^0, w^0)$ *labeled with* $l_G(v^0) \cap l_H(w^0)$ *is the root of* $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$*. Furthermore,* $p(v^0, w^0)$ *is set to* $(p(v^0), p(w^0))$ *and an* $\epsilon$*-cycle* $((v^0, w^0)\epsilon(v^0, w^0))$ *is obtained from the* $\epsilon$*-cycles* $(v^0\epsilon v^0)$*,* $(w^0\epsilon w^0)$*.*
*At each level* $k$ *such that* $0 < k \le min(|\mathcal{G}^\epsilon|, |\mathcal{H}^\epsilon|)$*,*

1. *For each* $r$*-successor (*$\forall r$*-successor)* $v^k_i$ *of* $v^{k-1}_i$ *in* $\mathcal{G}^\epsilon$ *and each* $r$*-successor (*$\forall r$*-successor)* $w^k_j$ *of* $w^{k-1}_j$ *in* $\mathcal{H}^\epsilon$ *such that* $(v^{k-1}_i, v^{k-1}_j)$ *is a node in* $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$*, we obtain a* $r$*-successor (*$\forall r$*-successor)* $(v^k_i, w^k_j)$ *of* $(v^{k-1}_i, v^{k-1}_j)$ *in* $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$*. Additionally, for each* $\epsilon$*-successor* $v^k_{l'}$ *of* $v^k_l$ *in* $\mathcal{G}^\epsilon$ *and for each* $\epsilon$*-successor* $w^k_{h'}$ *of* $w^k_h$ *in* $\mathcal{H}^\epsilon$ *such that* $(v^k_l, w^k_h)$ *is a node created in* $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$*, we obtain an* $\epsilon$*-successor* $(v^k_{l'}, w^k_{h'})$ *of* $(v^k_l, w^k_h)$ *in* $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$*. If* $l_G(v^k_i)$ *and* $l_H(w^k_j) \ne \{\bot\}$ *(or* $l_G(v^k_{l'})$ *and* $l_H(w^k_{h'}) \ne \{\bot\}$*) then the node* $(v^k_i, w^k_j)$ *(or* $(v^k_{l'}, w^k_{h'})$*) is labeled with* $l_G(v^k_i) \cap l_H(w^k_j)$ *(or* $l_G(v^k_{l'}) \cap l_H(w^k_{h'})$*). Furthermore,* $p(v^k_i, w^k_j)$ *(or* $p(v^k_{l'}, w^k_{h'})$*) is set to* $(p(v^k_i), p(w^k_j))$ *(or* $(p(v^k_{l'}), p(w^k_{h'}))$*).*
2. *For all nodes* $(v^k_0, w^k_1)$*, ... ,* $(v^k_0, w^k_n)$ *(or* $(v^k_1, w^k_0)$*, ...,* $(v^k_m, w^k_0)$*) obtained from 1. such that* $l_G(v^k_0) = \{\bot\}$ *(or* $l_H(w^k_0) = \{\bot\}$*), we obtain a subgraph* $(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon)((v^k_0, w^k_1),\ldots,(v^k_0, w^k_n))$ *(or* $(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon)((v^k_1, w^k_0),\ldots,(v^k_m, w^k_0))$*) from the induced subgraph* $\mathcal{H}^\epsilon(w^k_1, \ldots, w^k_n)$ *(or* $\mathcal{G}^\epsilon(v^k_1, \ldots, v^k_m)$*) by replacing each its node* $w$ *(or* $v$*) with* $(v^k_0, w)$ *(or* $(v, w^k_0)$*) where* $(v^k_0, w)$ *(or* $(v, w^k_0)$*) is labeled with* $l_H(w)$ *(or* $l_G(v)$*). Furthermore,* $p(v^k_0, w)$ *(or* $p(v, w^k_0)$*) is set to* $(p(v^k_0), p(w))$ *(or* $(p(v), p(w^k_0))$*).*

From concept descriptions (taken from [4]) given in Example 4, Figure 6 show how to compute the product graph of normalization graphs built from these concept descriptions and the description tree obtained from the product graph by applying Algorithm 1.

**Example 4** *Let*
$C_3 := \exists r.(\forall r.\forall r.P^0_3) \sqcap \exists r.(\forall r.\forall r.P^1_3) \sqcap \forall r.(\exists r.\forall r.P^0_2 \sqcap \exists r.\forall r.P^1_2 \sqcap \forall r.(\exists r.P^0_1 \sqcap \exists r.P^1_1))$
$D_3 := \exists r.\exists r.\exists r.(P^0_1 \sqcap P^1_1 \sqcap P^0_2 \sqcap P^1_2 \sqcap P^0_3 \sqcap P^1_3)$
*where* $P^i_j \in N_C$*,* $r \in N_R$*.*
*Note that* $\epsilon$*-cycles are useful for computing product graphs. For instance,* $\epsilon$*-cycle of node* $u_1$ *of tree* $\mathcal{G}^\epsilon_{D_3}$ *and* $\epsilon$*-edge* $(v_1\epsilon v_3)$ *of tree* $\mathcal{G}^\epsilon_{C_3}$ *yield* $\epsilon$*-edge* $((v_1, u_1)\epsilon(v_3, u_1))$ *of tree* $\mathcal{G}^\epsilon_{C_3} \times \mathcal{G}^\epsilon_{D_3}$*.*
*To simplify the presentation, the* $\epsilon$*-cycles are not added to the graphs in the figures.*

**Remark 6** *The size of the product graph of two normalization graphs* $\mathcal{G}^\epsilon = (V_G, E_G \cup E^\epsilon_G, l_G)$*,* $\mathcal{H}^\epsilon = (V_H, E_H \cup E^\epsilon_H, l_H)$ *is bounded by the product of the sizes of these normalization graphs. In fact, it holds that* $|V_{G\times H}| \le |V_G| \times |V_H|$*,* $|E_{G\times H}| \le |E_G| \times |E_H|$ *and* $|E^\epsilon_{G\times H}| \le |E^\epsilon_G| \times |E^\epsilon_H|$*.*

In the sequel, we will show that the level notion for product graphs can be defined from those for normalization graphs and the computation of the product of two normalization graphs preserves important properties of the neighbourhood notion. These notions guarantee that Algorithm 1 and Algorithm 2 can be applied to product graphs.
Each path from the root $(v^0, w^0)$ to a node $(v^k, w^k)$ of $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$ corresponds to two paths: the one is from $v^0$ to $v^k$ on $\mathcal{G}^\epsilon$ and the other is from $w^0$ to $w^k$ on $\mathcal{H}^\epsilon$. Moreover, the number of ordinary edges of all paths from $v^0$ $(w^0)$ to $v^k$ $(w^k)$ is constant since $\mathcal{G}^\epsilon$ and $\mathcal{H}^\epsilon$ are normalization graphs. Thus, the number of ordinary edges of all paths from $(v^0, w^0)$ to $(v^k, w^k)$ on $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$ is constant as well. This allows us to define the level of a node $(v^k, w^k)$ as the number of ordinary edges of all paths from the root to $(v^k, w^k)$. It means that two nodes corresponding to the endpoints of any $\epsilon$-edges are always at the same level.
Therefore, Definition 9 can be extended to n-ary product of graphs as follows:

$$\mathcal{G}^\epsilon_{C_1} \times \ldots \times \mathcal{G}^\epsilon_{C_n} := (\mathcal{G}^\epsilon_{C_1} \times \ldots \times \mathcal{G}^\epsilon_{C_{n-1}}) \times \mathcal{G}^\epsilon_{C_n}$$

**Definition 10** *We denote* $\mathcal{T}^{\mathcal{E}}$ *as the set containing all normalization graphs and product graphs i.e*

$$\mathcal{T}^{\mathcal{E}} := \bigcup_{n\ge 1}\{\mathcal{G}^\epsilon_{C_1} \times \ldots \times \mathcal{G}^\epsilon_{C_n} \mid C_1, \ldots, C_n \text{ are } \mathcal{ALE}\text{-concept descriptions}\}$$

We now clarify how the definition of neighbourhood (Definition 6) can be applied to product graphs. Similarly to $\forall r$-neighbourhoods in normalization graphs, the computation of the $\forall r$-neighbourhood at level $k$ of a $(k-1)$-neighbourhood in product graphs takes into account the set of

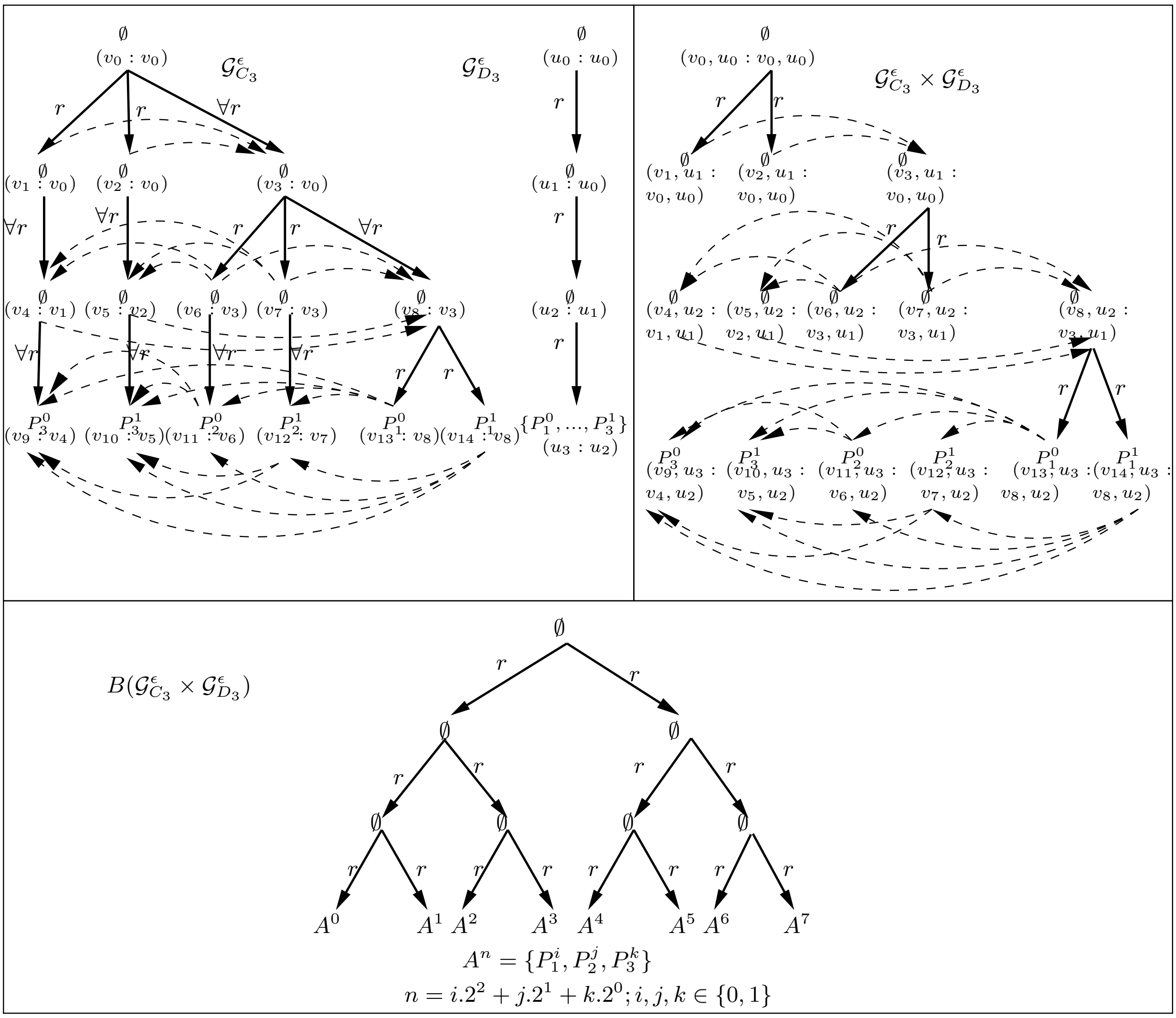


Figure 6. Product of normalization graphs

nodes $V^\epsilon$ in Definition 6 (neighbourhood). Differently from $\forall r$-neighbourhoods in normalization graphs where sets $V^\epsilon \neq \emptyset$ include only nodes whose label is equal to $\{\bot\}$, sets $V^\epsilon$ corresponding to $\forall r$-neighbourhoods in product graphs can contain nodes which have $r$-successors or $\forall r$-successors. More precisely,

**Lemma 4** *Let* $n_G^{k-1} = \{u_1, ..., u_m\}$ *and* $n_H^{k-1} = \{w_1, ..., w_n\}$ *be* $(k-1)$*-neighbourhoods respectively in* $\mathcal{G}^\epsilon$, $\mathcal{H}^\epsilon \in \mathcal{T}^\mathcal{E}$. *Let* $n_{G\times H}^{k-1}$ *be a* $(k-1)$*-neighbourhood in* $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$. *Assume that* $\{(u_1, w_1), ..., (u_m, w_n)\} \subseteq n_{G\times H}^{k-1}$ *and* $l_{G\times H}(u_i, w_j) = \emptyset$, $(u_i, w_j)$ *does not have any* $r$*-successor and* $\forall r$*-successor for all* $(u_i, w_j) \in n_{G\times H}^{k-1} \setminus \{(u_1, w_1), ..., (u_m, w_n)\}$. *It holds that there exist* $r$*-neighbourhoods* ($\forall r$*-neighbourhoods*) $n_G^k = \{v_1, ..., v_h\}$ *and* $n_H^k = \{z_1, ..., z_l\}$ *such that* $n_G^k \in N(n_G^{k-1})$ *and* $n_H^k \in N(n_H^{k-1})$ *iff there exists a* $r$*-neighbourhood* ($\forall r$*-neighbourhood*) $n_{G\times H}^k \in N(n_{G\times H}^{k-1})$ *such that* $\{(v_1, z_1), ..., (v_h, z_l)\} \subseteq n_{G\times H}^k$ *and* $l_{G\times H}(v_i, z_j) = \emptyset$, $(v_i, z_j)$ *does not have any* $r$*-successor and* $\forall r$*-successor for all* $(v_i, z_j) \in n_{G\times H}^k \setminus \{(v_1, z_1), ..., (v_h, z_l)\}$.

A proof of Lemma 4 can be found in Appendix. We now are ready to formulate and prove a theorem which establishes the relationship between the product of two graphs in $\mathcal{T}^\mathcal{E}$ and the product of two description trees (as defined in [3]) represented by these two graphs.

**Theorem 2** *Let $\mathcal{G}^\epsilon$, $\mathcal{H}^\epsilon \in \mathcal{T}^\mathcal{E}$. There exists an isomorphism between $\mathbf{B}(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon)$ and $\mathbf{B}(\mathcal{G}^\epsilon) \times \mathbf{B}(\mathcal{H}^\epsilon)$.*

A proof of Theorem 2 can be found in Appendix. This proof builds inductively on the depth of trees an isomorphism between the trees $\mathbf{B}(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon)$ and $\mathbf{B}(\mathcal{G}^\epsilon) \times \mathbf{B}(\mathcal{H}^\epsilon)$. In fact, assume that for each $(k-1)$-neighbourhood $n^{k-1}_{G\times H}$ in $(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon)$ we have two corresponding $(k-1)$-neighbourhoods $n^{k-1}_G$, $n^{k-1}_H$ on $\mathcal{G}^\epsilon$ and $\mathcal{H}^\epsilon$, respectively. The proof shows that $n^k_G = (u_1, ..., u_m)$, $n^k_H = (w_1, ..., w_n)$ are $k$-neighbourhoods respectively of $n^{k-1}_G$ and $n^{k-1}_H$ such that $\mathsf{label}(n^k_G) \neq \{\perp\}$, $\mathsf{label}(n^k_H) \neq \{\perp\}$ iff $n^k_{G\times H}$ is a $k$-neighbourhood of $n^{k-1}_{G\times H}$ such that $\{(u_1, w_1), ..., (u_m, w_n)\} \subseteq n^k_{G\times H}$ and $l_{G\times H}(v_i, z_j) = \emptyset$, $(v_i, z_j)$ does not have any $r$-successor and $\forall r$-successor for all $(v_i, z_j) \in n^k_{G\times H} \setminus \{(v_1, z_1), ..., (v_h, z_l)\}$. The proof of this claim is based heavily on Lemma 4. Additionally, if $\mathsf{label}(n^k_G) = \{\perp\}$ or $\mathsf{label}(n^k_H) = \{\perp\}$ then, Definition 9 yields that $(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon)(n^k_{G\times H})$ is equal to $\mathcal{H}^\epsilon(n^k_H)$ or $\mathcal{G}^\epsilon(n^k_G)$ (up to renaming nodes). Thus, $\mathbf{B}((\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon)(n^k_{G\times H}))$ is equal to $\mathbf{B}(\mathcal{H}^\epsilon(n^k_H))$ or $\mathbf{B}(\mathcal{G}^\epsilon(n^k_G))$. The construction of the isomorphism will be done by proving that $\mathsf{label}(n^k_G) = \{\perp\}$ and $\mathsf{label}(n^k_H) = \{\perp\}$ iff $\mathsf{label}(n^k_{G\times H}) = \{\perp\}$.

Proposition 1 yields that $\mathcal{G}_C$ and $\mathcal{G}_D$ are equal respectively to $\mathbf{B}(\mathcal{G}^\epsilon_C)$ and $\mathbf{B}(\mathcal{G}^\epsilon_D)$ (up to renaming nodes) and Proposition 2 shows that it is sufficient to use normalization graphs $\mathcal{G}^\epsilon_C$ , $\mathcal{G}^\epsilon_D$ rather than description trees $\mathcal{G}_C$ and $\mathcal{G}_D$ to decide subsumption between two $\mathcal{ALE}$-description concepts $C$ and $D$. Moreover, according to an important result in [3], the $lcs$ of $C$ and $D$ can be computed as the product $\mathcal{G}_C \times \mathcal{G}_D$. This result and Theorem 2 allow us to represent all $lcs$ as product graphs $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$ and decide subsumption between $lcs$ by manipulating directly the corresponding product graphs. Thus, we can define that the semantics of a graph $\mathcal{G}^\epsilon \in \mathcal{T}^\mathcal{E}$ is the semantics of the concept $C_{B(\mathcal{G}^\epsilon)}$ *i.e* for an interpretation $(\Delta, .^\mathcal{I})$, we define $(\mathcal{G}^\epsilon)^\mathcal{I} := (C_{B(\mathcal{G}^\epsilon)})^\mathcal{I}$. By consequent, we can talk about the subsumption, equivalence, $lcs$, *etc.* for all graphs $\mathcal{G}^\epsilon \in \mathcal{T}^\mathcal{E}$. The following result is a direct consequence of Theorem 2.

**Corollary 1** *Let $\mathcal{G}^\epsilon$, $\mathcal{H}^\epsilon \in \mathcal{T}^\mathcal{E}$. The least common subsumer of $\mathcal{G}^\epsilon$, $\mathcal{H}^\epsilon$ can be computed in polynomial time.*

Additionally, according to the definition of $lcs$, we have $C = lcs(C, \perp)$ for every $\mathcal{ALE}$-concept description $C$. Therefore, Proposition 2 can be generalized as follows.

**Proposition 3** *Let $\mathcal{G}^\epsilon$ and $\mathcal{H}^\epsilon$ be two product graphs corresponding to $lcs(C_1, C_2)$ and $lcs(D_1, D_2)$ where $C_1$, $C_2$, $D_1$ and $D_2$ are $\mathcal{ALE}$-concept descriptions i.e $\mathcal{G}^\epsilon = \mathcal{G}^\epsilon_{C_1} \times \mathcal{G}^\epsilon_{C_2}$ and $\mathcal{H}^\epsilon = \mathcal{G}^\epsilon_{D_1} \times \mathcal{G}^\epsilon_{D_2}$. Algorithm 2 applied to $\mathcal{G}^\epsilon$ and $\mathcal{H}^\epsilon$ can decide subsumption between $lcs(C_1, C_2)$ and $lcs(D_1, D_2)$ in polynomial space and exponential time.*

## 5. On the approximation $\mathcal{ALC}$-$\mathcal{ALE}$

In [1], a double exponential algorithm has been proposed for the approximation $\mathcal{ALC}$-$\mathcal{ALE}$. In this algorithm, the approximation is computed by using the $lcs$. A question left open by the authors concerns the existence of an *exponential algorithm* for computing the approximation. In the first attempt at finding an answer to this question, we hoped that if there is a method for obtaining a polynomial representation for the $lcs$, such a method may be applied for reducing the exponential blow-up caused by the distribution of disjunctions over conjunctions in the normalization for $\mathcal{ALC}$-concept descriptions. However, though the polynomial representation for the $lcs$ presented in Section 4 helps to reduce the size of the approximation, this representation does not allow for reducing the complexity class.

In this section, we formulate and prove a theorem which provides a tight lower bound of the size of the approximation $\mathcal{ALC}$-$\mathcal{ALE}$ in the ordinary representation.

**Theorem 3** *Let $C$ be an $\mathcal{ALC}$-concept description. The size of $approx_{\mathcal{ALE}}(C)$ may be double exponential in the size of $C$.*

The following proof of Theorem 3 uses some notions and the approximation algorithm described in [1].

Let $C$ is an $\mathcal{ALC}$-concept description where disjunction only occurs within value or existential restrictions. $Prim(C)$ denotes a set of all (negated)

**Algorithm 3.** $approx_{\mathcal{ALE}}(C)$ [1]
**Require:** $C$ is an $\mathcal{ALC}$-concept description in $\mathcal{ALC}$-normal form $C = C_1 \sqcup ... \sqcup C_n$.
**Ensure:** $approx_{\mathcal{ALE}}(C)$
  **if** $C \equiv \bot$ **then**
    **return** $\bot$;
  **end if**
  **if** $C \equiv \top$ **then**
    **return** $\top$;
  **else**
    **return**
$$\bigsqcap_{A \in \bigcap_{i=1}^{m} Prim(C_i)} A \sqcap \bigsqcap_{(C'_1,...,C'_m) \in Ex(C_1) \times ... \times Ex(C_m)} \{\exists r.lcs\{ approx_{\mathcal{ALE}}(C'_j \sqcap Val(C_j)) | 1 \le j \le m\}\} \sqcap \forall r.lcs\{approx_{\mathcal{ALE}}(Val(C_j)) | 1 \le j \le m\}$$
  **end if**

concept names occurring on the top-level conjunction of $C$ (the top-level conjunction is not wrapped within a value restriction or existential restriction). $Val(C)$ is the conjunction of all $C'$ occurring in value restrictions of form $\forall r.C'$ on top-level of $C$. If there is no value restriction on top-level of $C$ then $Val(C) = \top$. $Ex(C)$ is the set of all $C'$ occurring in existential restrictions of form $\exists r.C'$on top-level of $C$. The normal form of $C$ is defined as follows. Let $C$ be an $\mathcal{ALC}$-concept description and $C \not\equiv \top$, $C \not\equiv \bot$. $C$ is in $\mathcal{ALC}$-normal form iff $C$ is of the form $C_1 \sqcup ... \sqcup C_m$ such that
$C_i = \sqcap_{A \in Prim(Ci)} A \sqcap \bigsqcap_{C' \in Ex(Ci)} \exists r.C' \sqcap \forall r.Val(C_i)$, $\bot \sqsubset C_i$ where $C'$,$Val(C_i)$ are in $\mathcal{ALC}$-normal form.
If $C$ is in $\mathcal{ALC}$-normal form and $C \not\equiv \top$, $C \not\equiv \bot$, the approximation of $C$ can be computed by Algorithm 3 (the approximation algorithm in [1]). Considering the algorithm, the $\mathcal{ALC}$-normal form of an $\mathcal{ALC}$-concept description $C$ may contain $2^n$ disjuncts (assume that the initial form of $C$ is the conjunction of $n$ conjuncts and each conjunct is a binary disjunction). Furthermore, the value and existential restrictions in $approx_{\mathcal{ALE}}(C)$ may require to compute the $lcs$ of $2^n$ terms. If the $lcs$ under the existential restrictions do not subsume each other (absorption), the approximation may contain a double exponential number ($2^{2^n}$) of existential restrictions which do not subsume each other. This remark is useful for constructing the proof of Theorem 3.

We now characterize some properties that an $\mathcal{ALC}$-concept description $C$ leading to the exponential blow-up should satisfy:

(1) $C$ is the conjunction of $n$ conjuncts and each conjunct is a binary disjunction. Therefore, the $\mathcal{ALC}$-normal form of $C$ has $2^n$ disjuncts and each disjunct is the conjunction including $n$ conjuncts.

(2) From Algorithm 3, the approximation contains $2^{2^n}$ existential restrictions $\exists r.lcs\{E_{i1}, ..., E_{ik}\}$ where $k = 2^n$, $i \in \{1, .., 2^{2^n}\}$. Each $E_{ij}$ should be an existential restriction $\exists r.E_{ij}$ where $E_{ij}$ is conjunction of concept names belonging to $\{P^u_l, Q_v\}$ for $u, v \in \{1, 2\}$, $l \in \{1, ..., n\}$. Thus, each $lcs\{E_{i1}, ..., E_{ik}\}$ is conjunction of $\exists r.F^j_i$, and each $F^j_i$ is conjunction of concept names belonging to $\{P^u_r, Q_v\}$ for $u, v \in \{1, 2\}$, $r \in \{1, ..., n\}$. Each $F^j_i$ can be considered as a subset of $\{P^u_r, Q_v \mid u, v \in \{1, 2\}, r \in \{1, ..., n\}\}$.

(3) The essential property of $\exists r.lcs\{E_{i1}, ..., E_{ik}\}$ for $i \in \{1..2^{2^n}\}$, $k = 2^n$ to be guaranteed, is that they do not subsume each other. This means that for each pair of existential restrictions $\exists r.lcs\{E_{i1}, ..., E_{ik}\}$ and $\exists r.lcs\{E_{j1}, ..., E_{jk}\}$, $i, j \in \{1, ..2^{2^n}\}$, there exists a conjunct $\exists r.F^r_i$ of $\exists r.lcs\{E_{i1}, ..., E_{ik}\}$ such that $\exists r.F^r_i \not\sqsubseteq \exists r.F^s_j$ for all conjuncts $\exists r.F^s_j$ of $\exists r.lcs\{E_{j1}, ..., E_{jk}\}$, and vice versa.
The difficult in proving the property (3) is due the computing of $lcs\{E_{i1}, ..., E_{ik}\}$. This task may become easier if we partition existential restrictions obtained from $lcs\{E_{i1}, ..., E_{ik}\}$ into groups and identify "representative elements" of each group. Therefore, we only need to consider "representative elements" for deciding whether $lcs\{E_{i1}, ..., E_{ik}\}$ is absorbed by $lcs\{E_{j1}, ..., E_{jk}\}$.

The proof of Theorem 3 needs the following lemma.

**Lemma 5** *Let $(A^{i_1}_1, ..., A^{i_n}_n)$ be a n-dimension vector where $i_j \in \{1, 2\}$. Let $I$ be a bijection from $\{1, 2\} \times ... \times \{1, 2\}$ into $\{1, .., 2^n\}$ for numbering all vectors $\{(A^{i_1}_1, ..., A^{i_n}_n) \mid (i_1, ..., i_n) \in \{1, 2\} \times ... \times \{1, 2\}\}$. We have that each $(i_1, ..., i_n) \in \{1, 2\} \times ... \times \{1, 2\}$ determines uniquely $k \in \{1, .., 2^n\}$ such that $I(\bar{i}_1, ..., \bar{i}_n) = k$ where $\bar{i}_h \neq i_h$ for all $h \in \{1..n\}$.*

The proof of the lemma is trivial since $(\bar{i}_1, ..., \bar{i}_n) \in \{1, 2\} \times ... \times \{1, 2\}$ for each $(i_1, ..., i_n) \in \{1, 2\} \times ... \times \{1, 2\}$ and $I$ is a bijection.

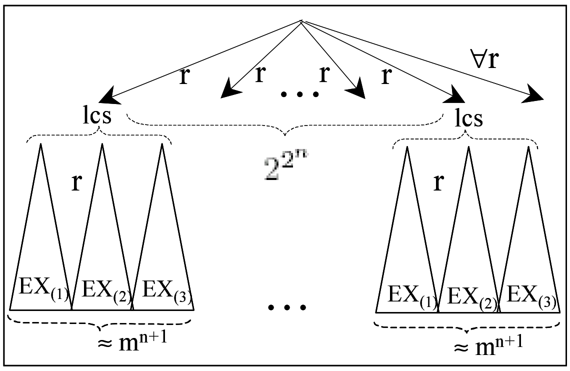


Figure 7. Double exponential $approx_{\mathcal{ALE}}(C)$

*Proof of Theorem* 3.

The theorem will be proven if there exists an $\mathcal{ALC}$-concept description $C$ such that the size of $approx_{\mathcal{ALE}}(C)$ is double exponential in the size of $C$ and $approx_{\mathcal{ALE}}(C)$ is irreducible. Let

$A_k^1 = \exists r.(P_k^1 \sqcap \bigsqcap_{i=1, i\neq k}^n (P_i^1 \sqcap P_i^2) \sqcap Q_1 \sqcap Q_2)$,
$A_k^2 = \exists r.(P_k^2 \sqcap \bigsqcap_{i=1, i\neq k}^n (P_i^1 \sqcap P_i^2) \sqcap Q_1 \sqcap Q_2)$
for $k \in \{1, ..., n\}$,

$B_1 = \exists r.(Q_1 \sqcap \bigsqcap_{i=1}^n (P_i^1 \sqcap P_i^2))$,
$B_2 = \exists r.(Q_2 \sqcap \bigsqcap_{i=1}^n (P_i^1 \sqcap P_i^2))$ where $P_k^i, Q_j \in N_C$, $r \in N_R$ for $i, j \in \{1, 2\}$, $k \in \{1, ..., n\}$.

Let $C$ be an $\mathcal{ALC}$-concept description:

$$C := \exists r.B_1 \sqcap \exists r.B_2 \sqcap \bigsqcap_{i=1}^n (\forall r.A_i^1 \sqcup \forall r.A_i^2)$$

We prove that the number of top-level existential restrictions of $approx_{\mathcal{ALE}}(C)$ is $2^{2^n}$ and these existential restrictions do not subsume each other.
The $\mathcal{ALC}$-normal form of $C$ is as follows:
$C \equiv C_1 \sqcup ... \sqcup C_m$ where
$C_i \equiv (\exists r.B_1 \sqcap \exists r.B_2 \sqcap \forall r.Val(C_i))$ and
$Val(C_i) = A_1^{j_2} \sqcap ... \sqcap A_n^{j_n}$, $(j_1, ..., j_n) \in (\{1, 2\} \times ... \times \{1, 2\})$.
According to Algorithm 3, we have:

$$approx_{\mathcal{ALE}}(C) = \bigsqcap_{(B_{i_1}, ..., B_{i_m}) \in (\{B_1, B_2\} \times ... \times \{B_1, B_2\})} \{\exists r.lcs\{(B_{i_j} \sqcap Val(C_j)) | 1 \leq j \leq m\}\} \sqcap \forall r.lcs\{Val(C_j) | 1 \leq j \leq m\} \text{ (*)}$$

Figure 7 shows $2^{2^n}$ existential restrictions on top-level of the expression (*). The expressions under these existential restrictions are $lcs$ and each one applies to $m = 2^n$ terms. We denote $E$ as the set of existential restrictions obtained from computing $lcs\{(B_{i_j} \sqcap Val(C_j)) | 1 \leq j \leq m\}$. According to the computation of the n-ary $lcs$, $E$ may contain $m^{n+1}$ existential restrictions. $E$ is partitioned into three subsets $EX_{(1)}$, $EX_{(2)}$, $EX_{(3)}$ as follows.
Since each tuple $(B_{i_1}, ..., B_{i_m}) \in \{B_1, B_2\} \times ... \times \{B_1, B_2\}$ determines a set $E$ of existential restrictions, we define a function $\mathcal{E}$ from the domain $\{(B_{i_1}, ..., B_{i_m}) | \ (B_{i_1}, ..., B_{i_m}) \in \{B_1, B_2\} \times ... \times \{B_1, B_2\}\}$ to the set of sets of existential restrictions obtained from the computing of $lcs\{(B_{i_j} \sqcap Val(C_j)) | 1 \leq j \leq m\}$ for all $(i_1, ..., i_m) \in \{1, 2\} \times ... \times \{1, 2\}$.
Each set $\mathcal{E}(X_1, ..., X_m)$ where $(X_1, ..., X_m) \in \{B_1, B_2\} \times ... \times \{B_1, B_2\}$ contains $m^{n+1}$ existential restrictions but some of them can be subsumed by others. In fact, $lcs\{(X_j \sqcap Val(C_j)) | 1 \leq j \leq m\}$ can be computed as the conjunction of $lcs\{E_{i_1}, ..., E_{i_m}\}$ where $E_{i_r} \in \{X_r\} \cup Val(C_r)$ for $r \in \{1, ..., m\}$. If $\{E_{i_1}, ..., E_{i_m}\} \subseteq \{E_{l_1}, ..., E_{l_m}\}$ then $lcs\{E_{i_1}, ..., E_{i_m}\} \sqsubseteq lcs\{E_{l_1}, ..., E_{l_m}\}$. Furthermore, we define a selection function
$\mathcal{S}(X_1, ..., X_m) := \{(E_{i_1}, ..., E_{i_m})\}$ where $E_{i_r} \in \{X_r\} \cup Val(C_r)$, $r \in \{1, ..., m\}$. This implies that $\mathcal{E}(X_1, ..., X_m) = \{lcs\{E_{i_1}, ..., E_{i_m}\} | \ (E_{i_1}, ..., E_{i_m}) \in \mathcal{S}(X_1, ..., X_m)\}$.
We will identify from all them the representative existential restrictions which form the three following subsets $EX_{(1)}$, $EX_{(2)}$ and $EX_{(3)}$ :

1. $EX_{(1)}(X_1, ..., X_m)$ is composed of the existential restrictions (of $\mathcal{E}(X_1, ..., X_m)$) that subsume the following existential restrictions: $lcs\{A_k^1, A_k^2\} \equiv \exists r.(Q_1 \sqcap Q_2 \sqcap \bigsqcap_{l=1, l\neq k}^n (P_l^1 \sqcap P_l^2))$ for $k \in \{1..n\}$.
   It is obvious that for each $k \in \{1..n\}$ there exists $(E_{i_1}, ..., E_{i_m}) \in \mathcal{S}(X_1, ..., X_m)$ such that $\{E_{i_1}, ..., E_{i_m}\} = \{A_k^1, A_k^2\}$ where $E_{i_j} = A_k^1 \in Val(C_j)$ or $E_{i_j} = A_k^2 \in Val(C_j)$ for all $j \in \{1..m\}$.
2. $EX_{(2)}(X_1, ..., X_m)$ is composed of the existential restrictions that subsume the following existential restrictions:
   $lcs\{B_i, B_j\} \equiv \exists r.(\bigsqcap_{k=1}^n (P_k^1 \sqcap P_k^2))$ if there exist $X_p, X_q \in \{X_1, ..., X_m\}$ and $X_p \neq X_q$, or $lcs\{B_i, B_i\} \equiv \exists r.(Q_i \sqcap \bigsqcap_{k=1}^n (P_k^1 \sqcap P_k^2))$ for $i \in \{1, 2\}$ if $X_1 = ... = X_m$.
   It is obvious that: $lcs\{B_i, B_j\} \in \mathcal{E}(X_1, ..., X_m)$ if there exist $X_p, X_q \in \{X_1, ..., X_m\}$, $X_p \neq X_q$. In fact, there exists $(E_{i_1}, ..., E_{i_m}) \in \mathcal{S}(X_1, ..., X_m)$ such that $\{E_{i_1}, ..., E_{i_m}\} =$

$\{B_i, B_j\}$ where $E_{i_r} = B_i \in \{X_r\} \cup Val(C_r)$ or $E_{i_r} = B_j \in \{X_r\} \cup Val(C_r)$ for all $r \in \{1..m\}$. Similarly, $lcs\{B_i, B_i\} \in \mathcal{E}(X_1, ..., X_m)$ if $X_1 = ... = X_m$.

3. $EX_{(3)}(X_1, ..., X_m)$ is composed of the existential restrictions that are *subsumed* by the following existential restrictions: $lcs\{X_k, A_1^{l_1}, A_2^{l_2}, ..., A_n^{l_n}\}$ where $(l_1, ..., l_n) \in \{1, 2\} \times ... \times \{1, 2\}$, $k = I(\bar{l}_1, ..., \bar{l}_n)$. The function $I$ is defined as follows: each conjunct $\exists r.lcs\{(X_j \sqcap Val(C_j))|1 \leq j \leq m\}$ on top-level of $approx_{\mathcal{ALE}}(C)$ where $Val(C_j) = A_1^{l_2} \sqcap ... \sqcap A_n^{l_n}$, determines $I(l_1, ..., l_n) = j$ where $j = (l_1 - 1).2^{n-1} + ... + (l_n - 1).2^0 + 1$ (the binary value of $(l_1, ..., l_n)$ plus 1). It is obvious that $I$ is a bijection. According to Lemma 5, $k = I(\bar{l}_1, ..., \bar{l}_n)$ is uniquely determined from $(l_1, ..., l_n)$ (**).
We have that $lcs\{X_k, A_1^{l_1}, A_2^{l_2}, ..., A_n^{l_n}\} \in \mathcal{E}(X_1, ..., X_m)$ for some $(l_1, ..., l_n) \in \{1, 2\} \times ... \times \{1, 2\}$, $k = I(\bar{l}_1, ..., \bar{l}_n)$. This is implied from Lemma 5 *i.e* for each $(l_1, ..., l_n) \in \{1, 2\} \times ... \times \{1, 2\}$ there exists $(E_{i_1}, ..., E_{i_m}) \in \mathcal{S}(X_1, ..., X_m)$ such that $\{E_{i_1}, ..., E_{i_m}\} = \{X_k, A_1^{l_1}, A_2^{l_2}, ..., A_n^{l_n}\}$ where $E_{i_k} = X_k$ for $k = I(\bar{l}_1, ..., \bar{l}_n)$ and $E_{i_r} = A_s^{l_s} \in Val(C_r)$ for some $s \in \{1..n\}$ and $r \neq k$.

To show $\mathcal{E}(X_1, ..., X_m) = EX_{(1)}(X_1, ..., X_m) \cup EX_{(2)}(X_1, ..., X_m) \cup EX_{(3)}(X_1, ..., X_m)$, we only need to show that if $e \in \mathcal{E}(X_1, ..., X_m)$, $e \notin EX_{(2)}(X_1, ..., X_m)$ and $e \notin EX_{(1)}(X_1, ..., X_m)$, then $e \in EX_{(3)}(X_1, ..., X_m)$.
Indeed, if $e \notin EX_{(2)}(X_1, ..., X_m)$ and $e \notin EX_{(1)}(X_1, ..., X_m)$ then $e$ has to be of the form $e = lcs\{B_{i_k}, A'_1, ..., A'_p\}$ such that $B_{i_j} \in \{B_1, B_2\}$, $\{A'_1, ..., A'_p\} \subseteq \{A_1^{j_1}, ..., A_n^{j_n}\}$ for some $(j_1, ..., j_n) \in \{1, 2\} \times ... \times \{1, 2\}$. Note that if there exists $h \in \{1..n\}$ such that $A_h^1, A_h^2 \in \{A'_1, ..., A'_p\}$ then $e \in EX_{(1)}(X_1, ..., X_m)$. Moreover, we have $B_{i_k} = X_k$ where $k = I(\bar{j}_1, ..., \bar{j}_n)$ since $A_h^{j_h} \notin Val(C_k)$ for all $h \in \{1..n\}$.
This means that if for all $(E_{i_1}, ..., E_{i_m}) \in \mathcal{S}(X_1, ..., X_m)$ such that $A_h^1$ and $A_h^2 \notin \{E_{i_1}, ..., E_{i_m}\}$ for all $h \in \{1..n\}$ but $B_1$ or $B_2 \in \{E_{i_1}, ..., E_{i_m}\}$ and $A_{s_1}^{i_{s_1}}, ..., A_{s_p}^{i_{s_p}} \in \{E_{i_1}, ..., E_{i_m}\}$ for $s_1, ..., s_p \in \{1..n\}$, then $\{E_{i_1}, ..., E_{i_m}\} = \{X_k, A_{s_1}^{i_{s_1}}, ..., A_{s_p}^{i_{s_p}}\}$ such that $\{A_{s_1}^{i_{s_1}}, ..., A_{s_p}^{i_{s_p}}\} \subseteq \{A_1^{j_1}, ..., A_n^{j_n}\}$ for some $(j_1, ..., j_n) \in \{1, 2\} \times ... \times \{1, 2\}$ and $E_{i_k} = X_k = B_1$ or $B_2$ for $k = I(\bar{j}_1, ..., \bar{j}_n)$.

Therefore, $e \sqsubseteq lcs\{B_{i_k}, A_1^{j_1}, ..., A_n^{j_n}\}$ and thus $e \in EX_{(3)}(X_1, ..., X_m)$.

We now prove that for each couple $(X_1, ..., X_m)$, $(Y_1, ..., Y_m) \in \{B_1, B_2\} \times ... \times \{B_1, B_2\}$, $(X_1, ..., X_m) \neq (Y_1, ..., Y_m)$, the following properties are verified:
$lcs\{(X_u \sqcap Val(C_u))|1 \leq u \leq m\} \not\sqsubseteq lcs\{(Y_v \sqcap Val(C_v))|1 \leq v \leq m\}$, and
$lcs\ (Y_v \sqcap Val(C_v))|1 \leq v \leq m\} \not\sqsubseteq lcs\{(X_u \sqcap Val(C_u))|1 \leq u \leq m\}$.
According to the definition of function $\mathcal{E}$, these properties are reformulated as follows:

There exists $e' \in \mathcal{E}(Y_1, ..., Y_m)$ such that $e'' \not\sqsubseteq e'$ for all $e'' \in \mathcal{E}(X_1, ..., X_m)$ and,
there exists $e'' \in \mathcal{E}(X_1, ..., X_m)$ such that $e' \not\sqsubseteq e''$ for all $e' \in \mathcal{E}(Y_1, ..., Y_m)$.

Owing to the partition of $\mathcal{E}$ into the subsets $EX_{(1)}$, $EX_{(2)}$, $EX_{(3)}$, considering only representative existential restrictions of the subsets is enough to prove the properties above.

Let $(X_1, ..., X_m), (Y_1, ..., Y_m) \in \{B_1, B_2\} \times ... \times \{B_1, B_2\}$ and $k_0 \in \{1, .., m\}$ such that $X_{k_0} \neq Y_{k_0}$. Without loss of generality, assume that:
$Y_{k_0} = B_1 = \exists r.(Q_1 \sqcap \bigsqcap_{l=1}^{n}(P_l^1 \sqcap P_l^2))$ and $X_{k_0} = B_2 = \exists r.(Q_2 \sqcap \bigsqcap_{l=1}^{n}(P_l^1 \sqcap P_l^2))$. We pick $e'' = \exists r.(Q_2 \sqcap P_1^{j_1} \sqcap ... \sqcap P_n^{j_n})$ from $EX_{(3)}(X_1, ..., X_m)$ where $\exists r.(Q_2 \sqcap P_1^{j_1} \sqcap ... \sqcap P_n^{j_n}) = lcs\{X_{k_0}, A_1^{j_1}, ..., A_n^{j_n}\}$ and $k_0 = I(\bar{j}_1, ..., \bar{j}_n)$ (function $I$ is defined above (**)). First, show that $e' \not\sqsubseteq e''$ for all $e' \in E(Y_1, ..., Y_m)$.

- $e' \not\sqsubseteq e''$ for all $e' \in EX_{(1)}(Y_1, ..., Y_m)$ since $\{Q_2, P_1^{j_1}, ..., P_n^{j_n}\} \not\subseteq \{Q_1, Q_2\} \cup \bigcup_{l=1, l \neq h}^{n}\{P_l^1, P_l^2\}$ for $h \in \{1..n\}$.
- $e' \not\sqsubseteq e''$ for all $e' \in EX_{(2)}(Y_1, ..., Y_m)$ since $\{Q_2, P_1^{j_1}, ..., P_n^{j_n}\} \not\subseteq \bigcup_{l=1}^{n}\{P_l^1, P_l^2\}$ and $\{Q_2, P_1^{j_1}, ..., P_n^{j_n}\} \not\subseteq \{Q_1\} \cup \bigcup_{l=1}^{n}\{P_l^1, P_l^2\}$. Note that $(Y_1, ..., Y_m) \neq (B_2, ..., B_2)$ since $Y_{k_0} = B_1$.
- $e' \not\sqsubseteq e''$ for all $e' \in EX_{(3)}(Y_1, ..., Y_m)$, $e' \notin EX_{(1)}(Y_1, ..., Y_m)$ and $e' \notin EX_{(2)}(Y_1, ..., Y_m)$. Indeed, $e'$ can be written as follows: $e' = lcs\{B_{i_h}, A_{s_1}^{i_{s_1}}, ..., A_{s_p}^{i_{s_p}}\}$ such that $\{A_{s_1}^{i_{s_1}}, ..., A_{s_p}^{i_{s_p}}\} \subseteq \{A_1^{l_1}, ..., A_n^{l_n}\}$ for some $(l_1, ..., l_n) \in \{1, 2\} \times ... \times \{1, 2\}$ and $B_{i_h} = Y_h$ where $h = I(\bar{l}_1, ..., \bar{l}_n)$. This means that there exists $(E_{i_1}, ..., E_{i_m}) \in \mathcal{S}(Y_1, ..., Y_m)$ such that $\{E_{i_1}, ..., E_{i_m}\} = \{Y_h, A_{s_1}^{i_{s_1}}, ..., A_{s_p}^{i_{s_p}}\}$ where $\{A_{s_1}^{i_{s_1}}, ..., A_{s_p}^{i_{s_p}}\} \subseteq$

$\{A_1^{j_1},...,A_n^{j_n}\}$ for some $(l_1,...,l_n) \in \{1,2\} \times ... \times \{1,2\}$ and $E_{i_h} = Y_h \in \{Y_h\} \cup Val(C_h)$ for $h = I(\bar{l}_1,...,\bar{l}_n)$.
Assume that $h = k_0$. We have $E_{i_h} = E_{i_{k_0}} = Y_{k_0} \neq X_{k_0}$ where $Y_{k_0} = B_1 = \exists r.(Q_1 \sqcap \bigsqcap_{l=1}^{n}(P_l^1 \sqcap P_l^2))$, $X_{k_0} = B_2 = \exists r.(Q_2 \sqcap \bigsqcap_{l=1}^{n}(P_l^1 \sqcap P_l^2))$. Hence, $e''$ contains $Q_2$ but $e'$ does not contains $Q_2$. Thus, $e' \not\sqsubseteq e''$.
Assume that $h \neq k_0$. If $E_{i_{k_0}} = Y_{k_0} = B_1$ then, according the argument above, $e' \not\sqsubseteq e''$. Otherwise, $E_{i_{k_0}} = A_r^{\bar{j}_r} \in \{A_1^{\bar{j}_r},...,A_n^{\bar{j}_r}\}$ and $A_r^{\bar{j}_r} \in \{A_{s_1}^{i_{s_1}}$,...,$A_{s_p}^{i_{s_p}}\}$ where $k_0 = I(\bar{j}_1,...,\bar{j}_n)$. Since $e''= lcs\{X_{k_0}, A_1^{j_1},...,A_n^{j_n}\}$ hence $e''$ contains $P_r^{j_r}$ for $j_r \in (j_1,...,j_n)$. On the other hand, since $e' = lcs\{Y_h, A_{s_1}^{i_{s_1}},...,A_{s_p}^{i_{s_p}}\}$ and $A_r^{\bar{j}_r} \in \{A_{s_1}^{i_{s_1}},...,A_{s_p}^{i_{s_p}}\}$ hence $e'$ contains $P_r^{\bar{j}_r}$ but not $P_r^{j_r}$. Thus, $e' \not\sqsubseteq e''$.

Similarly, we can show that there exists $e' \in E(Y_1,...,Y_m)$ such that $e'' \not\sqsubseteq e'$ for all $e'' \in E(X_1,...,X_m)$. To do it, pick $e' = \exists r.(Q_1 \sqcap P_1^{j_1} \sqcap ... \sqcap P_n^{j_n})$ from $E_{(3)}(Y_1,...,Y_m)$ where $\exists r.(Q_1 \sqcap P_1^{j_1} \sqcap ... \sqcap P_n^{j_n}) = lcs\{Y_k, A_1^{j_1},...,A_n^{j_n}\}$ and $k = I(\bar{j}_1,...,\bar{j}_n)$. We proceed in the same way as above.

It remains to be proven that there does not exist any $\mathcal{ALE}$-concept description $D$ such that $D \equiv approx_{ALE}(C)$ and the number of existential restrictions in $D$ (as conjuncts on top-level) is smaller than $2^{2^n}$. Assume that there exists such a concept description $D$. Since $C \sqsubseteq D$, the height of the description tree $\mathcal{G}(D)$ is not greater than 2. Furthermore, there exist existential restrictions $\exists r.C_1, \exists r.C_2$ where $C_1, C_2 \in Ex(approx_{ALE}(C))$ and an existential restriction $\exists r.D_1$ where $D_1 \in Ex(D)$ such that $D_1 \equiv C_1$, $D_1 \equiv C_2$. This implies that $C_1 \sqsubseteq C_2$, which contradicts the property of $approx_{ALE}(C)$ shown above. ■

**Remark 7** *Algorithm 3 yields immediately a* normalized *$\mathcal{ALE}$-concept description the number of existential restrictions on top-level of which may be double exponential. The fact is that the algorithm may generate non-collapsible $2^{2^n}$ existential restrictions from $2^n$ disjuncts on top-level of the $\mathcal{ALC}$-form normal of $C$ (as constructed in the proof) cannot be explained by the interaction between value and existential restrictions. It means that the compact representation introduced in Section 3 cannot help to reduce the size of the obtained approximation. Hence, the question raised is whether this exponential blow-up is specific to the approximation computation.*

**Algorithm 4.** $approx_{\mathcal{ALE}}(C)$
**Require:** $C$ is an $\mathcal{ALC}$-concept description in $\mathcal{ALC}$-normal form $C = C_1 \sqcup ... \sqcup C_n$.
**Ensure:** $approx_{\mathcal{ALE}}(C)$
  **if** $C \equiv \bot$ **then**
    **return** $\bot$;
  **end if**
  **if** $C \equiv \top$ **then**
    **return** $\top$;
  **end if**
  **if** $n = 1$ **then**
    **return** $\bigsqcap_{A \in prim(C_1)} A \sqcap$
      $\bigsqcap_{C' \in ex(C_1)} \exists r.approx_{\mathcal{ALE}}(C' \sqcap val(C_1)) \sqcap$
        $\forall r.approx_{\mathcal{ALE}}(val(C_1))$;
  **else**
    **return**
      lcs$\{approx_{\mathcal{ALE}}(C_1)$,..., $approx_{\mathcal{ALE}}(C_n)\}$
  **end if**

The remainder of this section will show that the exponential blow-up caused by the approximation computation results from the computing of the n-ary $lcs$ of $n$ $\mathcal{ALE}$-concept descriptions.
First, we need the following proposition for this purpose.

**Proposition 4** *Let $C = C_1 \sqcup ... \sqcup C_n$ be an $\mathcal{ALC}$-concept description where $\bot \sqsubset C_1,...,C_n$. The approximation of $C$ by $\mathcal{ALE}$-concept description can be computed as follows:*

$$approx_{\mathcal{ALE}}(C) \equiv \text{lcs}\{approx_{\mathcal{ALE}}(C_1), \; ... \; , approx_{\mathcal{ALE}}(C_n)\}$$

A proof of Proposition 4 can be found in Appendix.

Algorithm 4 is a direct consequence of Algorithm 3 and Proposition 4.
Algorithms 3, 4 provide two methods to compute the approximation. This allows us to conclude that a double exponential number of existential restrictions occurring on the top-level of the approximation obtained from Algorithm 3 is due to the computing of the $lcs$ of an exponential number of concept description.
More concretely, Algorithms 3 and 4 establish the following equivalence for $approx_{\mathcal{ALE}}(C)$ ($C$ is constructed in the proof of Theorem 3)

$$\bigsqcap_{(B_{i_1},...,B_{i_m})\in(\{B_1,B_2\}\times...\times\{B_1,B_2\})} \{\exists r.lcs\{ (B_{i_j} \sqcap Val(C_j))|1 \le j \le m\}\} \sqcap \forall r.lcs\{Val(C_j)|1 \le j \le m\}$$

$$\equiv$$

$$lcs\{\exists r.B_1 \sqcap \exists r.B_2 \sqcap \forall r.Val(C_1), \ldots , \exists r.B_1 \sqcap \exists r.B_2 \sqcap \forall r.Val(C_m)\}$$

Note that the left side can be obtained from the right side by computing directly the *lcs*.
The computing of *lcs* in the right side requires a normalization. If the rules in Definition 2 are used for the normalization, the size of the normalized concept descriptions increases polynomially. Thus, the exponential blow-up of the computing of *lcs* in this case is not due to the interaction between value and existential restrictions. This explains why the compact representation presented in Section 3 does not help to avoid the exponential blow-up. This result is compatible with the result shown in [3], which states that an exponential blow-up may occur for the computing of $lcs\{C_1,...,C_n\}$ where $C_i$ are $\mathcal{EL}$-concept descriptions *i.e* no normalization is necessary.

## 6. Conclusion and future work

We have presented a specific data structure, called *normalization graph*, for representing $\mathcal{ALE}$-concept descriptions. This data structure can represent the *lcs* of two $\mathcal{ALE}$-concept descriptions in a polynomial space. We have proposed an algorithm polynomial in space and exponential in time for deciding subsumption between $\mathcal{ALE}$-concept descriptions including *lcs*. This result allows us to add to a reasoner a procedure for treating *lcs* without increasing the complexity of subsumption inference in time and space.
This paper has shown that the size of the approximation $\mathcal{ALC}$-$\mathcal{ALE}$ in the compact representation, and thus, in the ordinary representation may be double exponential. This result together with double exponential complexity of Algorithm 3, as shown in [1], allows us to conclude that lower and upper bounds for the size of the approximation $\mathcal{ALC}$-$\mathcal{ALE}$ in the compact representation, and thus, in the ordinary representation are double exponential. This gives a *partial* answer to the question left open by the authors in [1]. What we can affirm from the results of the present paper is that there does not exist any exponential algorithm for computing the approximation $\mathcal{ALC}$-$\mathcal{ALE}$ in *the ordinary representation*. This affirmation does not mean that there does not exist any exponential algorithm for computing the approximation $\mathcal{ALC}$-$\mathcal{ALE}$. We may obtain a positive answer to this question if there exists a special representation for $\mathcal{ALE}$-concept descriptions, which enables to express the approximation $\mathcal{ALC}$-$\mathcal{ALE}$ in an exponential space.
Our method is based heavily on the characterization of subsumption by homomorphisms between description trees presented in [3]. This characterization that is extended to normalization graphs helps to avoid exponential blow-up of the size of the binary *lcs*, but does not allow to avoid double exponential size of the approximations. The computation performed in Section 5 shows that the double exponential blow-up in the approximation algorithm comes from the following two sources: i) distribution of conjunctions over disjunctions (shown in [8]), ii) exponential size of $lcs\{C_1, ..., C_n\}$. This explains why normalization graphs which compact only the normalization are not sufficient for avoiding the double exponential blow-up in the approximation algorithm. However, as illustrated in Figure 6, a very compact normalization graphs (polynomial) may replace a complete binary tree for representing an $\mathcal{ALE}$-concept description. This could provide an idea for finding a reduction limit of representations for $\mathcal{ALE}$-concept descriptions.
The work in [11] has given a double exponential algorithm for computing the *lcs* in the logic $\mathcal{ALEN}$. This yields a double exponential upper bound for the size of the *lcs* of two $\mathcal{ALEN}$-concept descriptions. Hence, a natural question raised is whether description $\epsilon$-trees preserve their properties in more expressive Description Logics, for example, $\mathcal{ALEN}$. This question deserves to be studied in future work.


## Acknowledgements

The authors wish to thank the anonymous reviewers of an earlier version of this paper for comments and suggestions that have helped us to significantly improve the quality as well as the readability of this paper.

## References


[1] S. Brandt, R. Küsters and A.-Y Turhan. Approximation and Difference in Description Logics. In D. Fensel, D. McGuiness, and M.-A Williams, eds. *Proc. of KR-2002,* pages 203-214, San Francisco, CA, 2002. Morgan Kaufmann.

[2] S.Brandt, R. Küsters and A.-Y Turhan. Approximating $\mathcal{ALCN}$-concept description*s. Proc. of DL2002.*

[3] F. Baader, R. Küsters and R. Molitor. Computing Least Common Subsumer in Description Logics with Existential Restrictions. In T. Deen, ed. *Proc. of IJCAI-99.* pages 96-101, Stockholm, Sweden, 1999, Morgan Kaufmann.

[4] F. Baader and A.-Y Turhan. On the problem of Computing Small Representations of Least Common Subsumers. In *Proc. of KI2002.* Lecture Notes in Artificial Intelligence. Aachen, Germany, 2002.

[5] F. Baader, R. Molitor and R. Tobies. Tractable and Decidable Fragments of Conceptual Graphs. In W. Cyre and W. Tepfenhart, eds. *Proc. of ICCS99.* Lecture Notes in Computer Science. Springer-Verlag, 1999.

[6] F. M. Donini, B. Hollunder, M. Lenzerini, A. M. Spaccamela, D. Nardi and W. Nutt. The complexity of existential quantification in concept languages. *Artificial Intelligence* (2), 309-327, 1992.

[7] M. Schmidt-Schauß and G. Smolka. Attribute concept descriptions with complements. *Artificial Intelligence (48), 1-26, 1991.*

[8] B. Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence* (43), 235-249, 1990.

[9] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider. The Description Logic Handbook: Theory, Implementation and Applications. *Cambridge Press 2003.*

[10] I. Horrocks, P.F. Patel-Schneider and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics.* 1(1):7-26, 2003.

[11] R. Küsters and R. Molitor. Computing Least Common Subsumers in $\mathcal{ALEN}$. In B. Nebel, ed., *Proc. of IJCAI-2001.* pages 219-224, 2001, Morgan Kaufmann.

[12] S. Brandt, R. Küsters and A.Y. Turhan. Extensions of Non-standard inferences to Description Logics with Transitive Roles. In M. Vardi and A. Voronkov, eds, *Proc. of LPAR03*, Lecture Notes in Artificial Intelligence, 2003, Springer-Verlag.


## Appendix

Proofs of Propositions and Theorems

**Lemma** 1 *Let $\mathcal{G}^\epsilon(C) = (V, E \cup E^\epsilon, l)$ be an $\epsilon$-tree. A node $v_1^l \in V$ has a* q-clash $[v_1^l, .., v_q^l]$, *$q \in \{1,2,3\}$ such that $\{v_1^l, .., v_q^l\} \subseteq n^l$ where $n^l$ is a $\forall r$-neighbourhood iff one of the three following conditions is satisfied:*

1. *there exists $v_i^l \in n^l$ such that $l(v_i^l) = \{\bot\}$;*
2. *there exists $v_i^l, v_j^l \in n^l$ such that $v_i^l, v_j^l$ are $\forall r$-successors, $(v_i^l \epsilon v_j^l) \in E^\epsilon$ and there exists $P \in N_C$ such that $P, \neg P \in l(v_i^l) \cup l(v_j^l)$.*
3. 
   (a) *there exist nodes $v_i^k, v_j^k \in V$ $(l < k)$ such that $(v_i^k \epsilon v_j^k) \in E^\epsilon$. Moreover, if $v_i^k \neq v_j^k$ then there exists $P \in N_C$ such that $P, \neg P \in l(v_i^k) \cup l(v_j^k)$. If $v_i^k = v_j^k$ then $l(v_i^k) = \{\bot\}$;*

   (b) *there exist neighbourhoods $n^l$, $n^{l+1} \in N(n^l)$, ... , $n^{k-1} \in N(n^{k-2})$, $n^k \in N(n^{k-1})$ and r-edges $(v_1^{m-1} r v_2^m) \in E$ such that $v_1^m, v_2^m \in n^m$ for all $l < m < k$, $(v_1^{k-1} r v_2^k) \in E$, $v_1^l \in n^l$ and $v_i^k, v_j^k, v_2^k \in n^k$.*

*Proof.*
2. "If-direction".
Assume that there exists $v_i^l, v_j^l \in n^l$ such that $(v_i^l \epsilon v_j^l) \in E^\epsilon$. Assume that $v_i^l = v_j^l$, $l(v_i^l) = \{\bot\}$. According to Definition 7 (clash), we obtain that there exists a 1-clash $[v_i^l]$. Since $C$ is in weak normal form, $v_i^l$ is a $\forall r$-successor. According to Definition 6 (neighbourhood), there exists a $\forall r$-neigbourhood $n^l$ such that $v_i^l \in n^l$.
Assume that $v_i^l \neq v_j^l$ , $v_i^l, v_j^l$ are $\forall r$-successors and there exist $(v_i^l \epsilon v_j^l)$ and $P \in N_C$ such that $P, \neg P \in l(v_i^l) \cup l(v_j^l)$. According to Definition 7 (clash), there exists a 2-clash $[v_i^l, v_j^l]$. According to Definition 6 (neighbourhood), there exists a $\forall r$-neighbourhood $n^l$ such that $v_i^l, v_j^l \in n^l$.
Let $n^l$, $n^{l+1} \in N(n^l)$,..., $n^{k-1} \in N(n^{k-2})$, $n^k \in N(n^{k-1})$ be neighbourhoods on the tree $\mathcal{G}^\epsilon(C)$ such that $l < k$, $n^l$ is a $\forall r$-neighbourhood, $v_i^k, v_j^k \in n^k$, $(v_i^k \epsilon v_j^k) \in E^\epsilon$ and $P, \neg P \in l(v_i^k) \cup l(v_j^k)$, $P \in N_C$ (or $\bot \in l(v_i^k) \cup l(v_j^k)$). Let $(v_1^{m-1} r v_2^m)$ be $r$-edges such that $v_1^m, v_2^m \in n^m$ for all $l < m < k$ and $v_1^l \in n^l$, $v_2^k \in n^k$, $(v_1^{k-1} r v_2^k) \in E$. The definition of neighbourhood yields that each $m$-neighbourhood contains the unique $r$-successor $v_2^m$ for all $l < m \leq k$. Moreover, since $v_i^k, v_j^k, v_2^k \in n^k$, $(v_1^{k-1} r v_2^k) \in E$, $v_1^{k-1} \in n^{k-1}$ and $n^k \in N(n^{k-1})$, we have that $p(v_i^k)$, $p(v_j^k) \in n^{k-1}$. Similarly, if $p^{k-m}(v_i^k)$, $p^{k-m}(v_j^k)$, $v_2^m \in n^m$, $(v_1^{m-1} r v_2^m) \in E$, $v_1^{m-1} \in n^{m-1}$ and $n^m \in N(n^{m-1})$, we have that $p^{k-m+1}(v_i^k)$, $p^{k-m+1}(v_j^k) \in n^{m-1}$ for all $l < m < k$. Hence, there exist $\epsilon$-edges $(v_2^m \epsilon v_1^m)$, $(v_2^m \epsilon p^{k-m}(v_i^k))$, $(v_2^m \epsilon p^{k-m}(v_j^k)) \in E^\epsilon$ for all $l <$

$m < k$ (it is possible that $v_2^m = v_1^m$ or $v_2^m = p^{k-m}(v_i^k)$ or $v_2^m = p^{k-m}(v_j^k)$) and $(v_2^k \epsilon v_i^k)$, $(v_2^k \epsilon v_j^k)$ $\in E^\epsilon$. Thus, $p^{k-m}(v_i^k)$, $p^{k-m}(v_j^k) \in n^m$ for all $l \leq m \leq k$, $v_1^l \in n^l$ and $v_2^k \in n^k$.
Assume that $v_i^k = v_2^k$. According to Definition 7 (clash), we obtain that there exists a 1-clash $[v_1^l]$ or 2-clash $[v_1^l, p^{k-l}(v_j^k)]$ where $n_1^l$, $p^{k-l}(v_j^k) \in n^l$.
Assume that $v_i^k \neq v_2^k$. According to Definition 7 (clash), we obtain that there exists a 1-clash $[v_1^l]$, 2-clash $[v_1^l, p^{k-l}(v_i^k)]$ or 3-clash $[v_1^l, p^{k-l}(v_i^k), p^{k-l}(v_j^k)]$ where $n_1^l$, $p^{k-l}(v_i^k)$, $p^{k-l}(v_j^k)$ $\in n^l$.
We now prove the "only-if-direction".

According to Definition 7 (clash), a q-clash $[v_1^l, .., v_q^l]$ can be created i) from a node $v_i^k$ such that $l(v_i^l) = \{\perp\}$, ii) from two $\forall r$-successors $v_i^k, v_j^k$ $\in \{v_1^l, .., v_q^l\}$ such that $(v_i^l \epsilon v_j^l) \in E^\epsilon$; $P$, $\neg P \in l(v_i^l)$ $\cup$ $l(v_j^l)$, $P \in N_C$ $(v_i^l \neq v_j^l)$ iii) from nodes $v_i^k, v_j^k$ and a $r$-predecessor $v_2^k$ such that $(v_2^k \epsilon v_i^k)$, $(v_2^k \epsilon v_j^k)$ $(v_i^k \epsilon v_j^k) \in E^\epsilon$ and $P$, $\neg P \in l(v_i^k) \cup l(v_j^k)$, $P \in N_C$ (or $\perp \in l(v_i^k) \cup l(v_j^k)$) . We consider the case iii).

First, we show that there exists a $l$-neighbourhood $n^l$ such that $v_1^l$, $v_i^l$, $v_j^l \in n^l$ where $v_i^l = p^{k-l}(v_i^k)$ and $v_j^l = p^{k-l}(v_j^k)$. Since $\mathcal{G}^\epsilon(C)$ without $\epsilon$-edges is a tree, hence there exists a node $v^c$ such that $v^c = p^{l-c}(v_1^l) = p^{l-c}(v_i^l) = p^{l-c}(v_j^l)$. Moreover, since there exist $\epsilon$-edges which connect all pairs of nodes $v_i^l, v_j^l, v_1^l$ ( $(v_2^{l+1} \epsilon p^{k-l-1}(v_i^k))$, $(v_2^{l+1} \epsilon p^{k-l-1}(v_j^k))$, $(p^{k-l-1}(v_i^k) \epsilon p^{k-l-1}(v_j^k)) \in E^\epsilon$), there exist $\epsilon$-edges between $p^{l-m}(v_1^l)$, $p^{l-m}(v_i^l)$ and $p^{l-m}(v_j^l)$ for all $c \leq m \leq l$, i.e by the definition of neighbourhood, there exists at most a $r$-edge between nodes $p^{l-m+1}(v_i^l)$, $p^{l-m+1}(v_j^l)$, $p^{l-m+1}(v_1^l)$ and $p^{l-m}(v_i^l)$, $p^{l-m}(v_j^l)$, $p^{l-m}(v_1^l)$ for all $c < m \leq l$. This yields that there exist neighbourhoods $n^c$, $n^{c+1}$ $\in N(n^c)$,...,$n^l$ $\in N(n^{l-1})$ such that $v^c \in n^c$ and $p^{l-m}(v_i^l)$, $p^{l-m}(v_j^l)$, $p^{l-m}(v_1^l) \in n^m$ for all $c \leq m \leq l$. Thus, $v_i^l, v_j^l, v_1^l \in n^l$ where $n^l$ is a $l$-neighbourhood (*). Furthermore, if $v_i^l, v_j^l, v_1^l$ are $\forall r$-successors then $n^l$ is a $\forall r$-neighbourhood.

By the construction of the q-clash $[v_1^l, .., v_q^l]$, there exist $r$-edges $(v_1^{m-1} r v_2^m)$ for all $l < m \leq k$ such that $(v_2^m \epsilon p^{k-m}(v_i^k))$, $(v_2^m \epsilon p^{k-m}(v_j^k))$, $(v_2^m \epsilon v_1^m)$ $\in E^\epsilon$. We define neighbourhoods $n^l$, $n^{l+1} \in N(n^l)$, ... , $n^{k-1} \in N(n^{k-2})$, $n^k \in N(n^{k-1})$ where $\{v_1^l, .., v_q^l\} \subseteq n^l$, $\{v_2^k, v_i^k, v_j^k\} \subseteq n^k$ as follows:

Let $n^l$ be a $l$-neighbourhood such that $p^{k-l}(v_i^k)$, $p^{k-l}(v_j^k)$, $v_1^l \in n^l$ (Note that $p^{k-l}(v_i^k)$, $p^{k-l}(v_j^k)$, $v_1^l \in \{v_1^l, .., v_q^l\}$) . There exists such a $n^l$ according to (*). Since $(v_2^{l+1} \epsilon p^{k-l-1}(v_i^k))$, $(v_2^{l+1} \epsilon p^{k-l-1}(v_j^k))$, $(p^{k-l-1}(v_i^k) \epsilon p^{k-l-1}(v_j^k)) \in E^\epsilon$ (Definition 7), we have that $v_2^{l+1}$ is the unique $r$-successor in the nodes $p^{k-l-1}(v_i^k)$, $p^{k-l-1}(v_j^k)$, $v_2^{l+1}$. Hence, we can define a $(l+1)$-neighbourhood $n^{l+1}$ such that $p^{k-l-1}(v_i^k)$, $p^{k-l-1}(v_j^k)$, $v_2^{l+1} \in n^{l+1}$ and $n^{l+1} \in N(n^l)$. Similarly, if $p^{k-m}(v_i^k)$, $p^{k-m}(v_j^k)$, $v_2^m \in n^m$, we can define a $(m+1)$-neighbourhood $n^{m+1}$ such that $p^{k-m-1}(v_i^k)$, $p^{k-m-1}(v_j^k)$, $v_2^{m+1} \in n^{m+1}$ and $n^{m+1} \in N(n^m)$ for all $l < m < k$. Note that $p^0(v_i^k)$ $= v_i^k$ and $p^0(v_j^k) = v_j^k$. ■

**Lemma 2** *Let $\mathcal{G}^\epsilon(C) = (V, E \cup E^\epsilon, l)$ be an $\epsilon$-tree ($v^0$ is its root) and $\mathcal{G}_C^\epsilon = (V', E' \cup E'^\epsilon, l')$ be its normalization graph. Let $n'^k$ be a $r$-neighbourhood ($\forall r$-neighbourhood) in $\mathcal{G}_C^\epsilon$.*

1. label$(n'^k) \neq \{\perp\}$ *iff*
   (a) *$n'^k$ is the 0-neighbourhood and $v^0$ does not have any* clash*, or*
   (b) *there exists a $r$-neighbourhood ($\forall r$-neighbourhood) $n^k$ in $\mathcal{G}^\epsilon(C)$ such that $n^k = n'^k$ $\cap$ $V$ and* label$(n^k)$ = label$(n'^k)$.
2. label$(n'^k) = \{\perp\}$ *iff*
   (a) *$n'^k$ is the 0-neighbourhood and $v^0$ has a* 1-clash $[v^0]$*, or*
   (b) *there exist a* q-clash$[v_1^k, .., v_q^k]$ *such that $\{v_1^k, .., v_q^k\} \subseteq n^k$, $n^k = V^\forall(n'^{k-1}) \cap V$ and $n'^k \in N(n'^{k-1})$ where $n^k$ is a $\forall r$-neighbourhood in $\mathcal{G}^\epsilon(C)$.*

*Proof.*
If $k = 0$ then, according to Definition 8 (normalization graph), $n'^0 = n^0 = \{v^0\}$. If label$(n'^0) \neq$ $\{\perp\}$ iff $l'(v^0) \neq \{\perp\}$. This implies that $l(v^0) \neq$ $\{\perp\}$ and $l(v^0)$ is not modified by the normalization (normalization graph) and thus $(v^0)$ does not contain any clash. In addition, $l'(v^0) = \{\perp\}$ iff $l(v^0) =$ $\{\perp\}$ or $l(v^0)$ is modified by the normalization (normalization graph). This implies that there exists a 1-clash $[v^0]$.
Assume that $n^k = n'^k \cap V$, $k \geq 0$ and label$(n^k)$ = label$(n'^k) \neq \{\perp\}$ (*). Let $n'^{k+1}$ be a $k$-neighbourhood of $n'^k$ in $\mathcal{G}_C^\epsilon$.
1 Assume that label$(n'^{k+1}) \neq \{\perp\}$. The definition of function label yields that $l(v'^{k+1}) \neq \{\perp\}$ for all $v'^{k+1} \in n'^{k+1}$.
Assume that $n'^{k+1}$ is the $\forall r$-neighbourhood of $n'^k$. From Definition 6 (neighbourhood), there exists

$(w^k \forall r w^{k+1}) \in E'$ such that $w^k \in n'^k$ and $w^{k+1} \in n'^{k+1}$. If $w^k \in n^k$ then $w^{k+1} \in n^{k+1}$ and thus there exists the $\forall r$-neighbourhood $n^{k+1}$ of $n^k$ in $\mathcal{G}^\epsilon(C)$. If $w^k \notin n^k$ then the normalization (normalization graph) yields that there exist $w'^k \in n^k$, $w'^{k+1} \in n^{k+1}$ such that $(w'^k \forall r w'^{k+1}) \in E$. This implies that there exists the $\forall r$-neighbourhood $n^{k+1}$ of $n^k$ in $\mathcal{G}^\epsilon(C)$ if there exists $n'^{k+1}$.
In addition, if $n'^{k+1} = V^\forall(n'^k)$ then $V^\forall(n^k) \subseteq V^\forall(n'^k)$ (since $n^k = n'^k \cap V$) and $l(v'^{k+1}) = \emptyset$ for all $v'^{k+1} \in n'^{k+1}$ and $v'^{k+1} \notin n^{k+1}$ since Definition 6 (neighbourhood) adds only $\forall r$-successors $v$ where $l'(v) = \emptyset$. If $n'^{k+1} = V^\epsilon(n'^k)$ then, according to Definition 6 (neighbourhood), $n'^{k+1}$ contains a node $v'^{k+1}$ such that $v'^{k+1} \notin V^\forall(n'^k)$, $p(v'^{k+1}) \in n'^k$, $(w^{k+1} \epsilon v'^{k+1}) \in E'^\epsilon$, $w^{k+1} \in V^\forall(n'^k)$. This implies that $v'^{k+1}$ is added by the normalization (Definition 6) and $l(v'^{k+1}) = \{\perp\}$ ($n'^{k+1}$ is neither a $\forall r$-successor nor a $r$-successor). Therefore, $\mathsf{label}(n'^{k+1}) = \{\perp\}$, which is a contradiction. Thus, $n^{k+1} = n'^{k+1} \cap V$ and $\mathsf{label}(n'^{k+1}) = \mathsf{label}(n^{k+1})$.
Let now $n'^{k+1}$ be a $r$-neighbourhood of $n'^k$ in $\mathcal{G}^\epsilon_C$ such that $(v^k r v^{k+1}) \in E$, $v^k \in n^k$, $v^{k+1} \in n^{k+1}$. There exists a $r$-neighbourhood $n^{k+1}$ of $n^k$ in $\mathcal{G}^\epsilon(C)$ such that $(v^k r v^{k+1}) \in E$ where $v^k \in n^k$, $v^{k+1} \in n^{k+1}$. It is obvious that $n^{k+1} = n'^{k+1} \cap V$. Assume that $v'^{k+1} \in n'^{k+1}$ such that $v'^{k+1} \notin n^{k+1}$. From Definition 8 (normalization graph), we have that $l'(v'^{k+1}) = \emptyset$. This implies that $\mathsf{label}(n'^{k+1}) = \mathsf{label}(n^{k+1})$.

Conversely, assume that $n^{k+1}$ is the $\forall r$-neighbourhood of $n^k$ in $\mathcal{G}^\epsilon(C)$. This implies that there exists the $\forall r$-neighbourhood $n'^{k+1}$ of $n'^k$ in $\mathcal{G}^\epsilon_C$ since $n^k = n'^k \cap V$. If $n'^{k+1} = V^\epsilon(n'^k)$ then $n^{k+1} = V^\forall(n^k) \not\subseteq V^\forall(n'^k)$. If $n'^{k+1} = V^\forall(n'^k)$ then $V^\forall(n^k) \subseteq V^\forall(n'^k)$ (since $n^k = n'^k \cap V$ ) and $l(v'^{k+1}) = \emptyset$ for all $v'^{k+1} \in n'^{k+1}$ and $v'^{k+1} \notin n^{k+1}$ since Definition 6 (neighbourhood) adds only $\forall r$-successors $v$ where $l'(v) = \emptyset$. Thus, $n^{k+1} = n'^{k+1} \cap V$ and $\mathsf{label}(n'^{k+1}) = \mathsf{label}(n^{k+1})$.
Now, assume that $n^{k+1}$ is a $r$-neighbourhood of $n^k$ in $\mathcal{G}^\epsilon(C)$. Since $n^k = n'^k \cap V$, there exists a $r$-neighbourhood $n'^{k+1}$ of $n'^k$ in $\mathcal{G}^\epsilon_C$ such that $v_i^{k+1}$ is a $r$-successor, $v_i^{k+1} \in n^{k+1}$ and $v_i^{k+1} \in n'^{k+1}$. It is obvious that $n^{k+1} = n'^{k+1} \cap V$. Assume that $v'^{k+1} \in n'^{k+1}$ such that $v'^{k+1} \notin n^{k+1}$. From Definition 8 (normalization graph), we have that $l'(v'^{k+1}) = \emptyset$. This implies that $\mathsf{label}(n'^{k+1}) = \mathsf{label}(n^{k+1})$.
By absurdity, assume that $\mathsf{label}(n'^{k+1}) = \{\perp\}$. The definition of function $\mathsf{label}$ yields that $l(w^{k+1}) = \{\perp\}$ for some $w^{k+1} \in n'^{k+1}$. By the normalization (normalization graph), we have $w^{k+1} \notin n^{k+1}$ and $w^{k+1}$ is not a $\forall r$-successor. Thus, $w^{k+1} \in V^\epsilon(n'^k) = n'^{k+1}$. By consequent, $n^{k+1} \not\subseteq n'^{k+1}$, which is a contradiction.
1.2 Assume that $\mathsf{label}(n'^{k+1}) = \{\perp\}$. The definition of function function $\mathsf{label}$ yields that $l(w^{k+1}) = \{\perp\}$ for some $w^{k+1} \in n'^{k+1}$.
Assume that $n'^{k+1}$ is a $r$-neighbourhood of $n'^k$. Let $v_i^{k+1}$ be the $r$-successor such that $v_i^{k+1} \in n'^{k+1}$. Let $n^{k+1}$ be the $r$-neighbourhood of $n^k$ such that $v_i^{k+1} \in n^{k+1}$. Similarly, if $n^k$ is a $r$-neighbourhood of $n^{k-1}$ we pick $n^{k-1}$. This process of construction is stopped if $n^l$ is a $\forall r$-neighbourhood. By Lemma 1, there exist a $\mathsf{q}$-$\mathsf{clash}$ $[v_1^l, .., v_q^l]$, $l < k$ such that $m^l$ is a $\forall r$-neighbourhood, $\{v_1^l, .., v_q^l\} \subseteq m^l$ (*). It is not possible that $l = 0$ since (*). On the other hand, by induction hypothesis, there exists neighbourhoods $n'^k$, $n'^{k-1}$,..., $n'^l$ such that $n^k = n'^k \cap V$ , $n^{k-1} = n'^{k-1} \cap V$,..., $n^l = n'^l \cap V$. From $n^l = n'^l \cap V$ ($n'^l$ may be a $r$-neighbourhood!) we have that $n^l \subseteq V^\forall(n'^{l-1})$ where $n'^l \in N(n'^{l-1})$. From induction hypothesis, we obtain that $\mathsf{label}(n'^l) = \{\perp\}$. This is a contradiction since there exists $n'^{l+1} \in N(n'^l)$.
Assume that $n'^{k+1}$ is the $\forall r$-neighbourhood of $n'^k$. Since $w^{k+1}$ is not a $\forall r$-successor and $w^{k+1}$ is added by the normalization (normalization graph), we have $n'^{k+1} = V^\epsilon(n'^k)$. We consider the following cases:
i) If $w^{k+1}$ comes from $\mathsf{1}$-$\mathsf{clash}$ $[v^{k+1}]$ then $(v^k \forall r v^{k+1}) \in E$, $(v^{k+1} \epsilon w^{k+1}) \in E'^\epsilon$ and $p(w^{k+1}) = v^k$. This implies that $v^{k+1} \in n^{k+1}$ where $n^{k+1}$ is the $\forall r$-neighbourhood of $n^k$. Since $n^k = n'^k \cap V$ hence $n^{k+1} \subseteq V^\forall(n'^k)$ .
ii) If $w^{k+1}$ comes from $\mathsf{2}$-$\mathsf{clash}$ $[v_1^{k+1}, v_2^{k+1}]$ then the normalization (normalization graph) yields that $(w_1^k \forall r w_1^{k+1}) \in E'$, $(w_1^{k+1} \epsilon w^{k+1}) \in E'^\epsilon$,
$p(w^{k+1}) = w_1^k$, $w_1^k \in n'^k$ and
$p^{k-m+2}(w_1^{k+1}) = p^{k-m+2}(v_j^{k+1})$,
$p^{k-m+1}(v_j^{k+1}) \neq v_i^m$, $v_j^{k+1} \in \{v_1^{k+1}, v_2^{k+1}\}$
where $m < k+1$ is the highest level from the root $v^0$ such that there exists a $r$-successor $v_i^m \in \{p^{k-m+1}(v_1^{k+1}),p^{k-m+1}(v_2^{k+1})\}$. Furthermore, there are $(v_1^k \forall r v_1^{k+1}) \in E$, $(v_i^m \epsilon p^{k-m+1}(w_1^{k+1})) \in E'^\epsilon$ where $p^{k-m+1}(v_i^{k+1}) = v_i^m$, $v_i^{k+1} \in \{v_1^{k+1}, v_2^{k+1}\}$ and $v_i^{k+1} \neq v_j^{k+1}$.
From the construction of nodes $w^m$, ..., $w_1^{k+1}$ by the normalization (normalization graph), we have that for each neighbourhood $n'^m$ if $p^{k-m+1}(w_1^{k+1}) \in n'^m$ then $v_i^m$, $p^{k-m+1}(w_1^{k+1})$, $p^{k-m+1}(v_j^{k+1}) \in n'^m$ since there exist $(v_i^m \epsilon p^{k-m+1}(w_1^{k+1}))$,

$(v_i^m \epsilon p^{k-m+1}(v_j^{k+1})) \in E'^{\epsilon}$ and $p^{k-m+2}(w_1^{k+1}) = p^{k-m+2}(v_j^{k+1})$. By consequent, for each neighbourhood $n'^k$ if $w_1^k \in n'^k$ then $p(v_i^{k+1})$, $p(w_1^{k+1}) \in n'^k$. Since $w_1^k \in n'^k$ hence $p(v_i^{k+1})$, $w_1^k \in n'^k$. Thus, $v_1^{k+1}$, $v_2^{k+1} \in V^{\forall}(n'^k)$. Furthermore, since $n^k = n'^k \cap V$ hence $n^{k+1} = V^{\forall}(n'^k) \cap V$ and thus $v_1^{k+1}$, $v_2^{k+1} \in n^{k+1}$ where $n^{k+1}$ is the $\forall r$-neighbourhood of $n^k$.

iii) If $w^{k+1}$ comes from 3-clash $[v_1^{k+1}, v_2^{k+1}, v_3^{k+1}]$ then the normalization (normalization graph) yields that $(w_1^k \forall r w_1^{k+1}) \in E'$, $(w_1^{k+1} \epsilon w^{k+1}) \in E'^{\epsilon}$,
$p(w^{k+1}) = p(v_l^{k+1})$ and $w_1^k$, $p(v_l^{k+1}) \in n'^k$, $v_l^{k+1} \in \{v_1^{k+1}, v_2^{k+1}, v_3^{k+1}\}$,
$p^{k-m+2}(w_1^{k+1}) = p^{k-m+2}(v_j^{k+1})$ ,
$p^{k-m+1}(v_j^{k+1}) \neq v_i^m$, $v_j^{k+1} \in \{v_1^{k+1}, v_2^{k+1}, v_3^{k+1}\}$,
$p^{k-m+1}(v_l^{k+1}) \neq v_i^m$ and
$p^{k-m+1}(v_l^{k+1}) \neq p^{k-m+1}(v_j^{k+1})$
where $m < k+1$ is the highest level from the root $v^0$ such that there exists a $r$-successor $v_i^m \in \{p^{k-m+1}(v_1^{k+1}), p^{k-m+1}(v_2^{k+1}), p^{k-m+1}(v_2^{k+1})\}$. Furthermore, there is $(v_1^k \forall r v_1^{k+1}) \in E$,
$(v_i^m \epsilon p^{k-m+1}(w_1^{k+1})) \in E'^{\epsilon}$ where $p^{k-m+1}(v_1^{k+1}) = v_i^m$, $v_i^{k+1} \in \{v_1^{k+1}, v_2^{k+1}, v_3^{k+1}\}$ and $v_i^{k+1} \neq v_j^{k+1} \neq v_l^{k+1}$.
From the construction of nodes $w^m$, ..., $w_1^{k+1}$ by the normalization (normalization graph), we have that for each neighbourhood $n'^m$ if $p^{k-m+1}(w_1^{k+1})$, $p^{k-m+1}(v_l^{k+1}) \in n'^m$ then $v_i^m$, $p^{k-m+1}(w_1^{k+1})$, $p^{k-m+1}(v_j^{k+1})$, $p^{k-m+1}(v_l^{k+1}) \in n'^m$ since there exist $(v_i^m \epsilon p^{k-m+1}(w_1^{k+1}))$, $(v_i^m \epsilon p^{k-m+1}(v_j^{k+1})) \in E'^{\epsilon}$ and $p^{k-m+2}(w_1^{k+1}) = p^{k-m+2}(v_j^{k+1})$. By consequent, for each neighbourhood $n'^k$ if $w_1^k$, $p(v_l^{k+1}) \in n'^k$ then $p(v_i^{k+1})$, $p(w_1^{k+1})$, $p(v_j^{k+1})$, $p(v_l^{k+1}) \in n'^k$. Since $w_1^k$, $p(v_l^{k+1}) \in n'^k$ hence $p(v_i^{k+1})$, $p(w_1^{k+1})$, $p(v_j^{k+1})$, $p(v_l^{k+1}) \in n'^k$. Thus, $v_1^{k+1}$, $v_2^{k+1} \in V^{\forall}(n'^k)$. Furthermore, since $n^k = n'^k \cap V$ hence $n^{k+1} = V^{\forall}(n'^k) \cap V$ and thus $v_1^{k+1}$, $v_2^{k+1}$, $v_3^{k+1} \in n^{k+1}$ where $n^{k+1}$ is the $\forall r$-neighbourhood of $n^k$.

Conversely, assume that $n^{k+1}$ is the $\forall r$-neighbourhood of $n^k$ in $\mathcal{G}^{\epsilon}(C)$ and there is a q-clash $[v_1^{k+1}, ..., v_q^{k+1}]$ such that $\{v_1^{k+1}, ..., v_q^{k+1}\} \subseteq n^{k+1}$ and $n^{k+1} \subseteq V^{\forall}(n'^k)$ where $n'^{k+1} \in N(n'^k)$. Let $w^{k+1}$ be the node added by the normalization (normalization graph) for q-clash $[v_1^{k+1}, ..., v_q^{k+1}]$ such that $\mathsf{label}(w^{k+1}) = \{\bot\}$. We have that $(w_1^k \forall r w_1^{k+1}) \in E'$, $(w_1^{k+1} \epsilon w^{k+1}) \in E'^{\epsilon}$. If $w_1^k \in n^k$ then $w_1^{k+1} \in n^{k+1}$ (which corresponds to 1-clash). This implies that $p(w^{k+1}) = w_1^k$ and $w_1^{k+1} \in V^{\forall}(n'^k)$ since $n^{k+1} \subseteq V^{\forall}(n'^k)$. The definition of neighbourhood yields that $n'^{k+1} = V^{\epsilon}(n'^k)$ and $w^{k+1} \in n'^{k+1}$. Therefore, $\mathsf{label}(n'^{k+1}) = \{\bot\}$. If $w_1^k \notin n^k$ then $w_1^{k+1} \notin n^{k+1}$ (which corresponds to 2-clash or 3-clash), then
i) Case 2-clash $[v_1^{k+1}, ..., v_q^{k+1}] = [v_1^{k+1}, v_2^{k+1}]$. According to the normalization (normalization graph), there exists a highest level $m < k+1$ and a $r$-successor $v_i^m$ such that $v_i^m = p^{k-m+1}(v_i^{k+1})$, $v_i^{k+1} \in \{v_1^{k+1}, v_2^{k+1}\}$, $w_1^m = p^{k-m+1}(w_1^{k+1})$, $(v_i^m \epsilon w_1^{k+1}) \in E'^{\epsilon}$, $p(w_1^m) = p^{k-m+2}(v_j^{k+1})$, $v_j^{k+1} \in \{v_1^{k+1}, v_2^{k+1}\} \setminus \{v_i^{k+1}\}$. From the construction of nodes $w^m$, ..., $w_1^{k+1}$ by the normalization (normalization graph), we have that for every neighbourhood $n'^m$ if $n'^m$ contains $v_i^m$ and $p^{k-m+1}(v_j^{k+1})$ then $n'^m$ contains also $w_1^m$ since there exist $(v_i^m \epsilon p^{k-m+1}(w_1^{k+1})) \in E'^{\epsilon}$ and $p^{k-m+2}(w_1^{k+1}) = p^{k-m+2}(v_j^{k+1})$. Similarly, for every neighbourhood $n'^k$ if $n'^k$ contains $p(v_i^{k+1})$ and $p(v_j^{k+1})$ then $n'^k$ contains also $w_1^k$. Since $n^k = n'^k \cap V$ hence $p(v_i^{k+1})$, $p(v_j^{k+1}) \in n'^k$ and thus $w_1^k \in n'^k$. The definition of neighbourhood yields that $n'^{k+1} = V^{\epsilon}(n'^k)$ and $w^{k+1} \in n'^{k+1}$. Therefore, $\mathsf{label}(n'^{k+1}) = \{\bot\}$.
ii) Case 3-clash $[v_1^{k+1}, ..., v_q^{k+1}] = [v_1^{k+1}, v_2^{k+1}, v_3^{k+1}]$. According to the normalization (normalization graph), there exists a highest level $m < k+1$ and a $r$-successor $v_i^m$ such that $v_i^m = p^{k-m+1}(v_i^{k+1})$, $v_i^{k+1} \in \{v_1^{k+1}, v_2^{k+1}, v_3^{k+1}\}$, $w_1^m = p^{k-m+1}(w_1^{k+1})$, $(v_i^m \epsilon w_1^{k+1}) \in E'^{\epsilon}$, $p(w_1^m) = p^{k-m+2}(v_j^{k+1})$, $v_j^{k+1} \in \{v_1^{k+1}, v_2^{k+1}, v_3^{k+1}\} \setminus \{v_i^{k+1}\}$ and $p(w^{k+1}) = p(v_l^{k+1})$, $v_l^{k+1} \in \{v_1^{k+1}, v_2^{k+1}, v_3^{k+1}\} \setminus \{v_i^{k+1}, v_j^{k+1}\}$. From the construction of nodes $w^m$, ..., $w_1^{k+1}$ by the normalization (normalization graph), we have that for every neighbourhood $n'^m$ if $n'^m$ contains $v_i^m$, $p^{k-m+1}(v_j^{k+1})$ and $p^{k-m+1}(v_l^{k+1})$ then $n'^m$ contains also $w_1^m$ since there exist $(v_i^m \epsilon p^{k-m+1}(w_1^{k+1})) \in E'^{\epsilon}$ and $p^{k-m+2}(w_1^{k+1}) = p^{k-m+2}(v_j^{k+1})$. Similarly, for every neighbourhood $n'^k$ if $n'^k$ contains $p(v_i^{k+1})$, $p(v_j^{k+1})$ and $p(v_l^{k+1})$ then $n'^k$ contains also $w_1^k$. Since $n^k = n'^k \cap V$ hence $p(v_i^{k+1})$, $p(v_j^{k+1})$, $p(v_l^{k+1}) \in n'^k$ and thus $w_1^k \in n'^k$. The definition of neighbourhood yields that $n'^{k+1} = V^{\epsilon}(n'^k)$ and $w^{k+1} \in n'^{k+1}$. Therefore, $\mathsf{label}(n'^{k+1}) = \{\bot\}$. ■

**Lemma** 3 *Let $C$ be an $\mathcal{ALE}$-concept description in the weak normal form. There exists an isomorphism between $\mathbf{B}(\mathcal{G}_C^{\epsilon})$ and the description tree $\mathcal{H}$ which is obtained from $\mathbf{B}(\mathcal{G}^{\epsilon}(C))=(V_3, E_3, z^0, l_3)$ by applying exhaustively the following rules ($p$ is the predecessor function of $\mathbf{B}(\mathcal{G}^{\epsilon}(C))$):*

1. $P$, $\neg P \in l_3(z)$, $P \in N_C \rightarrow$
   $l_3(z) := \{\perp\}$ *(rule 5g)*
2. $(zrz') \in E_3$, $\mathbf{B}(\mathcal{G}^\epsilon(C))(z') = \mathcal{G}(\perp) \rightarrow$
   $\mathbf{B}(\mathcal{G}^\epsilon(C))(z) := \mathcal{G}(\perp)$ *(rule 6g)*
3. $\perp \in l_3(z) \rightarrow$
   $\mathbf{B}(\mathcal{G}^\epsilon(C))(z) := \mathcal{G}(\perp)$ *(rule 7g)*

*Proof.* Let $\mathcal{G}(C) = (V, E, v^0, l)$, its $\epsilon$-tree $\mathcal{G}^\epsilon(C) = (V, E \cup E^\epsilon, l)$ and normalization graph $\mathcal{G}^\epsilon_C = (V', E' \cup E'^\epsilon, l')$. Let $\mathbf{B}(\mathcal{G}^\epsilon_C) = (V_1, E_1, u^0, l_1)$, $\mathcal{H}$=$(V_2, E_2, w^0, l_2)$ and $\mathbf{B}(\mathcal{G}^\epsilon(C)) = (V_3, E_3, z^0, l_3)$. We will show this lemma by using Lemma 2. First, we prove the following claim:
**Lemma** For each node $z^k \in V_3$ it holds that $z^k \in V_2$ and $l_2(z^k) = \{\perp\}$ iff

1. there does not exist any path composed of $r$-edges from $z^h$ to $z^l$ $(l < h)$ in $\mathbf{B}(\mathcal{G}^\epsilon(C))$ such that $\perp \in l_3(z^h)$ or $P, \neg P \in l_3(z^h)$, and $p^n(v^k)$ $=z^l$ for some $n > 0$; and
2. $P, \neg P \in l_3(z^k)$ for some $P \in N_C$, or there exists a path composed of $r$-edges from $z^h$ to $z^k$ $(k < h)$ in $\mathbf{B}(\mathcal{G}^\epsilon(C))$ such that $\perp \in l_3(z^h)$ or $P, \neg P \in l_3(z^h)$.

Condition 1. guarantees that $v^k \in V_2$. In fact, if there exist such a path then, from rules (5g), (6g), (7g), we have $l_2(z^h) = \{\perp\}$. Since $p^n(v^k)$ $=z^l$ hence $z^k$ is deleted by rule 6g) or 7g). Conversely, for all nodes $z^l \in V_3$ such that $p^n(v^k)$ $=z^l$ for some $n > 0$, if there does not exist any path composed of $r$-edges from $z^h$ to $z^l$ $(l < h)$ in $\mathbf{B}(\mathcal{G}^\epsilon(C))$ such that $\perp \in l_3(z^h)$ or $P, \neg P \in l_3(z^h)$ then $z^k$ is not deleted from $V_3$ by rules 6g) or 7g).
In addition, if $P, \neg P \in l_3(z^k)$ for some $P \in N_C$ then $l_2(z^k) = \{\perp\}$. Assume that there exists a path composed of $r$-edges from $z^l$ to $z^k$ $(k < l)$ in $\mathbf{B}(\mathcal{G}^\epsilon(C))$ such that $\perp \in l_3(z^h)$ or $P, \neg P \in l_3(z^h)$, and $p^n(v^k)$ $=z^l$ for some $n > 0$. From rules (5g), (6g), (7g), we have $l_2(z^k) = \{\perp\}$. Conversely, assume that $P, \neg P \notin l_3(z^k)$ for all $P \in N_C$ and $l_2(z^k)$ $= \{\perp\}$. This implies that $l_3(z^k)$ is modified by rules (6g), (7g). Thus, there exists a path composed of $r$-edges from $z^h$ to $z^k$ $(k < h)$ in $\mathbf{B}(\mathcal{G}^\epsilon(C))$ such that $\perp \in l_3(z^h)$ or $P, \neg P \in l_3(z^h)$.
To construct a bijection between $V_2$ and $V_1$ , we can construct a bijection $\phi$ between the sets of neighbourhoods in $\mathcal{G}^\epsilon_C$ and $\mathcal{G}^\epsilon(C)$. Note that neighbourhoods $n^k$ such that $\mathsf{label}(n^k) = \{\perp\}$ correspond to leaves of trees i.e $N(n^k) = \emptyset$. We set $\phi(n^0)$ $= m^0$ where $n^0$, $m^0$ are 0-neighbourhoods, respectively, in $\mathcal{G}^\epsilon_C$ and $\mathcal{G}^\epsilon(C)$. It is obvious that $\mathsf{label}(n^0)$ $= \{\perp\}$ iff $\mathsf{label}(m^0) = \{\perp\}$. In fact, $\mathsf{label}(m^0)$ $= \{\perp\}$, by Lemma above, iff $P, \neg P \in \mathsf{label}(m^0)$ for some $P \in N_C$, or there exist neighbourhoods $m^1 \in N(m^0)$,...,$m^l \in N(m^{l-1})$ in $\mathcal{G}^\epsilon(C)$ and $r$-edges $(v^0rv^1)$,...,$(v^{l-1}rv^l) \in E$, $v^1 \in m^1$, $v^2 \in m^2$,..., $v^l \in m^l$ such that $P$, $\neg P \in l(v^l_i) \cup l(v^l_j)$, $v^l_i, v^l_j \in m^l$, $P \in N_C$ $(v^l_i \neq v^l_j)$ or $l(v^l_i) = \{\perp\}$ $(v^l_i = v^l_j)$. By Lemma 1, this implies that there exists a $\mathsf{1\text{-}clash}$ $[v^0]$, $v^0 \in n^0$. By Lemma 3, $\mathsf{label}(n^0)$ $= \{\perp\}$ iff there exists $\mathsf{1\text{-}clash}$ $[v^0]$.

Assume that $\phi(n^k) = m^k$ and $\mathsf{label}(n^k) = \mathsf{label}(m^k) \neq \{\perp\}$. Let $n^{k+1} \in N(n^k)$ be a $r$-neighbourhood ($\forall r$-neighbourhood) in $\mathcal{G}^\epsilon_C$. Assume that $\mathsf{label}(n^{k+1}) \neq \{\perp\}$. By Lemma 3, there exists a $r$-neighbourhood ($\forall r$-neighbourhood) $m^{k+1}$ $\in N(m^k)$ in $\mathcal{G}^\epsilon(C)$ such that $m^{k+1} = n^{k+1} \cap V_3$, and there does not exist any $\mathsf{q\text{-}clash}$ $[v^{k+1}_1, ..., v^{k+1}_q]$ such that $\{v^{k+1}_1, ..., v^{k+1}_q\} \subseteq m^{k+1}$. Lemma 1 yields that there do not exist neighbourhoods $m^{k+2} \in N(m^{k+1})$,..., $m^{l+1} \in N(m^l)$ $(l > k+1)$ in $\mathcal{G}^\epsilon(C)$ and $r$-edges $(v^{k+1}rv^{k+2})$, ..., $(v^{l-1}rv^l) \in E$, $v^{k+1} \in m^{k+1}$, ..., $v^l \in m^l$ such that $P$, $\neg P \in l(v^l_i) \cup l(v^l_j)$, $v^l_i, v^l_j \in m^l$, $P \in N_C$ or $\{\perp\} = l(v^l_i)$ $(v^l_i = v^l_j)$. By Lemma above, we have that the label of the node that corresponds to $m^{k+1}$ is different from $\{\perp\}$ i.e $\mathsf{label}(m^{k+1}) \neq \{\perp\}$ and thus $\mathsf{label}(n^{k+1}) = \mathsf{label}(m^{k+1})$. Conversely, from this neighbourhood $m^{k+1}$ we can determine uniquely $n^{k+1} = V^\forall(n^k)$ if $m^{k+1}$ is a $\forall r$-neighbourhood (i.e $m^{k+1} = n^{k+1} \cap V_3$), or $n^{k+1} = \{v^{k+1}\} \cup \{V^\epsilon_{v^{k+1}}\}$, $m^{k+1} = n^{k+1} \cap V_3$ (notations in the definition of neighbourhood) where $v^{k+1} \in m^{k+1}$ is a $r$-successor if $m^{k+1}$ is a $r$-neighbourhood. Thus $\mathsf{label}(n^{k+1}) = \mathsf{label}(m^{k+1})$ $\neq \{\perp\}$. We can follows the schema : $\mathsf{label}(m^{k+1})$ $\neq \{\perp\}$ (by Lemma above) => no existence of sequence of neighbourhood (Lemma 1) => no existence of $\mathsf{clash}$ (Lemma 3) => $\mathsf{label}(n^{k+1}) = \mathsf{label}(m^{k+1}) \neq \{\perp\}$. Therefore, we set $\phi(n^{k+1}) = m^{k+1}$.

Assume that $\mathsf{label}(n^{k+1}) = \{\perp\}$. By Lemma 3, it is obvious that $n^{k+1}$ is a $\forall r$-neighbourhood and there exists a $\mathsf{q\text{-}clash}$ $[v^{k+1}_1, ..., v^{k+1}_q]$ such that $\{v^{k+1}_1, ..., v^{k+1}_q\} \subseteq V^\forall(n^k)$. By Lemma 1, there exist neighbourhoods $m^{k+2} \in N(m^{k+1})$,..., $m^{l+1} \in N(m^l)$ $(l > k+1)$ in $\mathcal{G}^\epsilon(C)$, $m^{k+1}$ $\subseteq \{v^{k+1}_1, ..., v^{k+1}_q\}$ and $r$-edges $(v^{k+1}rv^{k+2})$, ..., $(v^{l-1}rv^l) \in E$, $v^{k+1} \in m^{k+1}$, ..., $v^l \in m^l$ such that $P$, $\neg P \in l(v^l_i) \cup l(v^l_j)$, $v^l_i, v^l_j \in m^l$, $P \in N_C$ or $\{\perp\} = l(v^l_i)$ $(v^l_i = v^l_j)$. By Lemma above, we have $\mathsf{label}(m^{k+1}) = \{\perp\}$ and $m^{k+1}$ is a $\forall r$-neighbourhood. Conversely, from this neighbour-

hood $m^{k+1}$ we can determine uniquely $n^{k+1} = V^{\forall}(n^k)$, $m^{k+1} = n^{k+1} \cap V_3$. Thus $\mathsf{label}(n^{k+1}) = \mathsf{label}(m^{k+1}) = \{\bot\}$. We set $\phi(n^{k+1}) = m^{k+1}$.
By consequent, we have constructed an isomorphism $\phi$ between $\mathcal{H}$ and $\mathbf{B}(\mathcal{G}^{\epsilon}_C)$ ■

**Proposition** 1 *Let $C$ be an $\mathcal{ALE}$-concept description. There exists an isomorphism between $\mathbf{B}(\mathcal{G}^{\epsilon}_C)$ and $\mathcal{G}_C$.*
*Proof.* Assume that $C$ is in the weak normal form. Let $\mathcal{G}(C) = (V, E, v^0, l)$ and its $\epsilon$-tree $\mathcal{G}^{\epsilon}(C) = (V, E \cup E^{\epsilon}, l')$. Let $\mathbf{B}(\mathcal{G}^{\epsilon}(C)) = (V_1, E_1, u^0, l_1)$. Let $\mathcal{G}(C') = (V_2, E_2, w^0, l_2)$ be the description tree of the concept description $C'$ obtained from $C$ by applying rules 1, 2 in Definition 2. From Lemma 3, we only need to prove that there exists an isomorphism between the tree obtained by applying the rules in Lemma 3 to $\mathbf{B}(\mathcal{G}^{\epsilon}(C))$, and $\mathcal{G}_C$. First, we show that there exists an isomorphism between $\mathbf{B}(\mathcal{G}^{\epsilon}(C))$ and $\mathcal{G}(C')$.
We construct by induction on level $k$ ($0 \leq k \leq |\mathcal{G}(C)|$) a bijection $\phi$: $V_2 \to V_1$ such that $l_2(w^0) = l_2(u^0)$, $l_2(w) = l_1(\phi(w))$ for all $w \in V_2$, and $(\phi(w_1)e\phi(w_2)) \in E_1$ for all $(w_1ew_2) \in E_2$.

**Level** $k = 0$. Since $\mathcal{G}^{\epsilon}(C)$ has unique 0-neighbourhood $(v^0)$, we obtain the root $u^0$ of $\mathbf{B}(\mathcal{G}^{\epsilon}(C))$ where $l_1(u^0) = l'(v^0) = l(v^0)$. We obtain also the root $w^0$ of $\mathcal{G}(C')$ where $l_2(w^0) = l(v^0)$. We set $\phi(w^0) := u^0$.

**Level** $k > 0$. Let $w^k \in V_2$ be a node at level $k$ of $\mathcal{G}(C')$. We have that $w^k$ corresponds to a set of nodes $\{v^k_1, ..., v^k_m\}$, $v^k_1, ..., v^k_m \in V$ resulting from the normalization. Hence, $l_2(w^k) = \{l(v^k_1) \cup ... \cup l(v^k_m)\}$. By induction hypothesis, assume that $\phi(w^k) = u^k$ where the node $u^k$, which is obtained from executing Algorithm 1 for operator $\mathbf{B}$, corresponds to the $k$-neighbourhood $(v^k_1, ..., v^k_m)$ of $\mathcal{G}^{\epsilon}(C)$ and $l_1(u^k) = \mathsf{label}(u^k) = \{l(v^k_1) \cup ... \cup l(v^k_m)\}$ (Note that $\mathcal{G}(C')$ and $\mathcal{G}^{\epsilon}(C)$ share the set of nodes $V$). If there is not any confusion we write *neighbourhood* $u^k$ for *node* $u^k$. We consider the following two cases:

1. Let $v^{k+1}_1, ..., v^{k+1}_l$ be all $\forall r$-successors of the nodes $v^k_1, ..., v^k_m$ i.e $\{v^{k+1}_1, ..., v^{k+1}_l\} = V^{\forall}(u^k)$. The application of normalization rule 1 yields a $(k+1)$-node $w^{k+1} = \{v^{k+1}_1, ..., v^{k+1}_l\}$ of $\mathcal{G}(C')$ and a $\forall r$-edge that connects $w^k$ to $w^{k+1}$ *i.e* $(w^k \forall r w^{k+1}) \in E_2$. Let $u^{k+1}$ be the $\forall r$-neighbourhood of $u^k$ i.e $u^{k+1} = V^{\forall}(u^k)$. If $\bot \in \{l(v^{k+1}_1) \cup ... \cup l(v^{k+1}_l)\}$, we have that $\mathsf{label}(u^{k+1}) = \{\bot\}$ and $\bot \in l_2(w^{k+1})$. Otherwise, $\mathsf{label}(u^{k+1}) = \{l(v^{k+1}_1) \cup ... \cup l(v^{k+1}_l)\} = l_2(w^{k+1})$. Therefore, the unique $\forall r$-successor $w^{k+1}$ of $w^k$ corresponds to the unique $\forall r$-successor $u^{k+1}$ of $u^k$ and $l_2(w^{k+1}) = l_1(u^{k+1})$. We set $\phi(w^{k+1}) := u^{k+1}$.
2. Let $v^{k+1}_0$ be a $r$-successor of one of nodes $v^k_1, ..., v^k_m$ and $v^{k+1}_1, ..., v^{k+1}_l$ be all $\forall r$-successors of nodes $v^k_1, ..., v^k_m$. The application of the normalization rule 2 yields a $(k+1)$-node $w^{k+1} = \{v^{k+1}_0, v^{k+1}_1, ..., v^{k+1}_l\}$ of $\mathcal{G}(C')$ and a $r$-edge that connects $w^k$ to $w^{k+1}$ *i.e* $(w^k r w^{k+1}) \in E_2$. Let $u^{k+1}$ and $(u^k r u^{k+1})$ be a node and a $r$-edge that are generated from node $u^k$ by Algorithm 1.
   Since $v^{k+1}_i$ is either a $\forall r$-successor or a $r$-successor of one of nodes $v^k_1, ..., v^k_m$ for all $i \in \{0, .., l\}$, we have $p(v^{k+1}_i) \in \{v^k_1, ..., v^k_m\}$. Furthermore, since each node $v^k_j$ where $v^k_j \in \{v^k_1, ..., v^k_m\}$ is connected to a node $v^k_i \in \{v^k_1, ..., v^k_m\}$ by an $\epsilon$-edge, hence by Definition 5, two nodes $v^{k+1}_0, v^{k+1}_j$ where $v^{k+1}_j \in \{v^{k+1}_1, ..., v^{k+1}_l\}$ are connected by an $\epsilon$-edge. Hence, according to Algorithm 1 for operator $\mathbf{B}$, the node $u^{k+1}$ corresponds to $(k+1)$-neighbourhood $(v^{k+1}_0, v^{k+1}_1, ..., v^{k+1}_l)$. (Note that $V^{\epsilon}_{v^k_0} = \{v^{k+1}_0, v^{k+1}_1, ..., v^{k+1}_l\}$, $(v^k_0 r v^{k+1}_0) \in E$) and $l_1(u^{k+1}) = \mathsf{label}(u^{k+1})$. If $\bot \in \{l(v^{k+1}_0) \cup l(v^{k+1}_1) \cup ... \cup l(v^{k+1}_l)\}$, we have that $\mathsf{label}(u^{k+1}) = \{\bot\}$ and $\bot \in l_2(w^{k+1})$. Otherwise, $\mathsf{label}(u^{k+1}) = \{l(v^{k+1}_0) \cup l(v^{k+1}_1) \cup ... \cup l(v^{k+1}_l)\} = l_2(w^{k+1})$.
   Conversely, from the $(k+1)$-neighbourhood $\{v^{k+1}_0, v^{k+1}_1, ..., v^{k+1}_l\}$ we can show that $\mathcal{G}(C')$ has a node $w^{k+1} = \{v^{k+1}_0, v^{k+1}_1, ..., v^{k+1}_l\}$ and a $r$-edge which connects $w^k$ to $w^{k+1}$.
   We set $\phi(w^{k+1}) := u^{k+1}$.

We have constructed a bijection $\phi$ as specified above from tree $\mathbf{B}(\mathcal{G}^{\epsilon}(C))$ into the tree $\mathcal{G}(C')$. According to Lemma 3, there exists an isomorphism between $\mathbf{B}(\mathcal{G}^{\epsilon}_C)$ and $\mathcal{H}$ where the tree $\mathcal{H}$ is obtained from $\mathbf{B}(\mathcal{G}^{\epsilon}(C))$ by applying the rules in Lemma 3. Therefore, it is sufficient to prove that the description tree $\mathcal{G}_C$ can be obtained from the tree $\mathcal{G}(C')$ by applying the rules in Lemma 3 (which correspond to rules 5, 6, 7 in Definition 2). In fact, each application of rules 5, 6, 7 to $C'$ corresponds to each application of rules 5g, 6g, 7g to $\mathcal{G}(C')$ and conversely. Moreover, the application of rules 5, 6, 7 to $C'$ allows us to obtain the strong normal form of $C$ from which the description tree $\mathcal{G}_C$ is built. Let $\mathcal{G}'$ be the tree obtained by applying the rules 5g, 6g, 7g to $\mathcal{G}(C')$. Thus, $\mathcal{G}'$ is isomorph to $\mathcal{G}_C$. ■

**Proposition** 2 *Let $C$ and $D$ be $\mathcal{ALE}$-concept descriptions, and let $\mathcal{G}^\epsilon_C$ and $\mathcal{G}^\epsilon_D$ be their normalization graphs. Algorithm 2 applied to $\mathcal{G}^\epsilon_C$ and $\mathcal{G}^\epsilon_D$ can decide subsumption between $C$ and $D$ in polynomial space and exponential time.*

*Proof.*

According to Remark 1, the transformation from $\mathcal{ALE}$-concept descriptions into the corresponding $\epsilon$-trees takes a polynomial time in the size of input concept descriptions $C$ and $D$ (note that $C$ and $D$ must be transformed into weak normal form before building the corresponding $\epsilon$-trees $\mathcal{G}^\epsilon(C)$ and $\mathcal{G}^\epsilon(D)$). Additionally, Remark 3 shows that adding nodes for stocking $\mathsf{clashes}$ increases polynomially the size of $\epsilon$-trees. Thus, the size of normalization graphs $\mathcal{G}^\epsilon_C$ and $\mathcal{G}^\epsilon_D$ is polynomial in the size of $C$ and $D$.

Algorithm 2 checks the existence of a homomorphism between between two description trees $\mathbf{B}(\mathcal{G}^\epsilon)$, $\mathbf{B}(\mathcal{H}^\epsilon)$. According to Theorem 1, Algorithm 2 allows us to decide subsumption between $C$ and $D$.

According to Definition 6 (neighbourhood), the number of $(k+1)$-neighbourhoods generated from a $k$-neighbourhood is polynomial in the size of $\mathcal{H}^\epsilon$ (or $\mathcal{G}^\epsilon$). Furthermore, since the height of $\mathcal{H}^\epsilon$ (or $\mathcal{G}^\epsilon$) is bounded by the size of tree and visited branches can be freed, the algorithm needs a piece of memory polynomial in the size of $\mathcal{H}^\epsilon$ (or $\mathcal{G}^\epsilon$) to store the neighbourhoods along the path $(w_0, w_{k+1}, ..., w_n)$ from root $w_0$ to leaf $w_n$. These paths are built by inductive calls in the algorithm. This implies that the algorithm takes an exponential time (*cf.* Remark 5) and a polynomial space. ■

**Lemma** 4 *Let $n^{k-1}_G = \{u_1, ..., u_m\}$ and $n^{k-1}_H = \{w_1, ..., w_n\}$ be $(k-1)$-neighbourhoods respectively in $\mathcal{G}^\epsilon$, $\mathcal{H}^\epsilon \in \mathcal{T}^{\mathcal{E}}$. Let $n^{k-1}_{G\times H}$ be a $(k-1)$-neighbourhood in $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$. Assume that*
*$\{(u_1, w_1), ..., (u_m, w_n)\} \subseteq n^{k-1}_{G\times H}$ and*
*$l_{G\times H}(u_i, w_j) = \emptyset$, $(u_i, w_j)$ does not have any r-successor and $\forall$r-successor for all*
*$(u_i, w_j) \in n^{k-1}_{G\times H} \setminus \{(u_1, w_1), ..., (u_m, w_n)\}$.*
*It holds that there exist r-neighbourhoods ($\forall$r-neighbourhoods) $n^k_G = \{v_1, ..., v_h\}$ and $n^k_H = \{z_1, ..., z_l\}$ such that $n^k_G \in N(n^{k-1}_G)$ and $n^k_H \in N(n^{k-1}_H)$ iff there exists a r-neighbourhood ($\forall$r-neighbourhood) $n^k_{G\times H} \in N(n^{k-1}_{G\times H})$ such that*
*$\{(v_1, z_1), ..., (v_h, z_l)\} \subseteq n^k_{G\times H}$ and $l_{G\times H}(v_i, z_j) = \emptyset$, $(v_i, z_j)$ does not have any r-successor and $\forall$r-successor for all*
*$(v_i, z_j) \in n^k_{G\times H} \setminus \{(v_1, z_1), ..., (v_h, z_l)\}$.*
*Proof* . (see the proof of Theorem **2**)

**Theorem 2** *Let $\mathcal{G}^\epsilon, \mathcal{H}^\epsilon \in \mathcal{T}^{\mathcal{E}}$. There exists an isomorphism between $\mathbf{B}(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon)$ and $\mathbf{B}(\mathcal{G}^\epsilon) \times \mathbf{B}(\mathcal{H}^\epsilon)$.*

**Lemma A.** *Let $\{u_1, ..., u_m\}$ and $\{w_1, ..., w_n\}$ be k-neighbourhoods respectively on graphs $\mathcal{G}^\epsilon_1 = (V_1, E_1 \cup E^\epsilon_1, l_1)$ and $\mathcal{G}^\epsilon_2 = (V_2, E_2 \cup E^\epsilon_2, l_2)$. Let $\{(u_1, w_1), ..., (u_m, w_n)\}$ be the corresponding k-neighbourhood of the product graph $\mathcal{G}^\epsilon_1 \times \mathcal{G}^\epsilon_2$. If* $\mathsf{label}\{u_1, ..., u_m\} = \{\bot\}$ ($\mathsf{label}\{w_1, ..., w_n\} = \{\bot\}$) *then* $\mathsf{label}\{(u_1, w_1), ..., (u_m, w_n)\} = \mathsf{label}\{w_1, ..., w_n\}$ ($\mathsf{label}\{u_1, ..., u_m\}$). *Otherwise, it holds that*

$$\mathsf{label}\{(u_1, w_1), ..., (u_m, w_n)\} = \mathsf{label}\{u_1, ..., u_m\} \cap \mathsf{label}\{w_1, ..., w_n\}$$

*Proof of the lemma.* According to the definition of function $\mathsf{label}$, $\mathsf{label}\{(u_1, w_1), ..., (u_m, w_n)\} = \{\bot\}$ iff $\mathsf{label}(u_i, w_j) = \{\bot\}$ for some $i \in \{1, ..., m\}$, $j \in \{1, ..., n\}$. From this, Definition 9 (product) yields that $\mathsf{label}\{u_1, ..., u_m\} = \{\bot\}$ and $\mathsf{label}\{w_1, ..., w_n\} = \{\bot\}$. Assume that
$\mathsf{label}\{u_1, ..., u_m\} \neq \{\bot\}$ and $\mathsf{label}\{w_1, ..., w_n\} \neq \{\bot\}$. (if $\mathsf{label}\{u_1, ..., u_m\} = \{\bot\}$ or $\mathsf{label}\{w_1, ..., w_n\} = \{\bot\}$ the Lemma is obvious from Definition 9). We have that $\mathsf{label}\{(u_1, w_1), ..., (u_m, w_n)\} = l(u_1, w_1) \cup ... \cup l(u_m, w_n) = (l(u_1) \cap l(w_1)) \cup ... \cup (l(u_m) \cap l(w_n))$;
On the other hand, according to the definition of function $\mathsf{label}$, it is that $\mathsf{label}\{u_1, ..., u_m\} = l(u_1) \cup ... \cup l(u_m)$ and $\mathsf{label}\{w_1, ..., w_n\} = l(w_1) \cup ... \cup l(w_n)$. Therefore, $\mathsf{label}\{(u_1, w_1), ..., (u_m, w_n)\} = \mathsf{label}\{u_1, ..., u_m\} \cap \mathsf{label}\{w_1, ..., w_n\}$.

*Proof of the theorem.*
Let $\mathcal{G}^\epsilon = (V_G, E_G \cup E^\epsilon_G, l_G)$ and $\mathcal{H}^\epsilon = (V_H, E_H \cup E^\epsilon_H, l_H)$ and $v^0, w^0$ be the roots of $\mathcal{G}^\epsilon$, $\mathcal{H}^\epsilon$. We denote $|\mathcal{G}^\epsilon|$ as the depth of graph $\mathcal{G}^\epsilon$. Assume that $|\mathcal{G}^\epsilon| \leq |\mathcal{H}^\epsilon|$. We will construct by induction on the level of graph $\mathcal{G}^\epsilon$ an isomorphism $\phi$ from tree $\mathbf{B}(\mathcal{G}^\epsilon) \times \mathbf{B}(\mathcal{H}^\epsilon) = (V_2, E_2, z^0, l_2)$ to tree $\mathbf{B}(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon) = (V_1, E_1, x^0, l_1)$.

**Level** $k = 0$.
At level 0, since product $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$ has unique neighbourhood $\{(v^0, w^0)\}$ without outgoing or ingoing

$\epsilon$-edge (with the exception of $\epsilon$-cycle), $\mathbf{B}(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon)$ has the root $x^0 = (v^0, w^0)$. Similarly, since $\mathcal{G}^\epsilon$ has unique node $v^0$ without outgoing or ingoing $\epsilon$-edge (with the exception of $\epsilon$-cycle) and $\mathcal{H}^\epsilon$ has unique node $w^0$ without outgoing or ingoing $\epsilon$-edge at level 0 (with the exception of $\epsilon$-cycle), thus $\mathbf{B}(\mathcal{G}^\epsilon) \times \mathbf{B}(\mathcal{H}^\epsilon)$ has the root $z^0 = (v^0, w^0)$.

Assume that $l_G(v^0) = \{\perp\}$ ($l_H(w^0) = \{\perp\}$). From Definition 9 we have that $\mathbf{B}(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon) = \mathbf{B}(\mathcal{H}^\epsilon)$ ($\mathbf{B}(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon) = \mathbf{B}(\mathcal{G}^\epsilon)$). On the other hand, we also have $\mathbf{B}(\mathcal{G}^\epsilon) = \{v^0\}$ ($\mathbf{B}(\mathcal{H}^\epsilon) = \{w^0\}$) where $l(v^0) = \{\perp\}$ ($l(w^0) = \{\perp\}$) (Algorithm 1), and thus $\mathbf{B}(\mathcal{G}^\epsilon) \times \mathbf{B}(\mathcal{H}^\epsilon) = \mathbf{B}(\mathcal{H}^\epsilon)$ ($\mathbf{B}(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon) = \mathbf{B}(\mathcal{G}^\epsilon)$) . If $l_G(v^0) \neq \{\perp\}$, $l_1(x^0) = \mathsf{label}(v^0, w^0) = (l_G(v^0) \cap l_H(w^0))$ and $l_2(z^0) = (l_G(v^0) \cap l_H(w^0))$. Thus, we set $\phi(z^0) := x^0$.

**Level** $k > 0$

Let $m^{k-1} = \{u_1, ..., u_m\}$ be a $(k-1)$-neighbourhood of $\mathcal{G}^\epsilon$ and $n^{k-1} = \{w_1, ..., w_n\}$ be a $(k-1)$-neighbourhood of $\mathcal{H}^\epsilon$. Let $q^{k-1}$ be a $(k-1)$-neighbourhood of $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$ such that $\phi(m^{k-1}, n^{k-1}) = q^{k-1}$. If $\mathsf{label}(m^{k-1}) = \{\perp\}$ and $\mathsf{label}(n^{k-1}) = \{\perp\}$ then $\mathsf{label}(q^{k-1}) = \{\perp\}$ (since $\phi$ is an isomorphism) and $N^k(q^{k-1}) = N^k(m^{k-1}) = N^k(n^{k-1}) = \emptyset$.

Let $m^{k-1} \times n^{k-1} = \{(u_1, w_1), ..., (u_m, w_n)\}$ and we denote $(m^{k-1}, n^{k-1})$ as a node or a neighbourhood in $\mathbf{B}(\mathcal{G}^\epsilon) \times \mathbf{B}(\mathcal{H}^\epsilon)$. As induction hypothesis, we assume that

1. $m^{k-1} \times n^{k-1} \subseteq q^{k-1}$,
2. One of two following conditions is satisfied:
   (a) $\mathsf{label}(m^{k-1}, n^{k-1}) = \mathsf{label}(q^{k-1}) = \{\perp\}$; $q^{k-1} = V^\epsilon$; and for all $(u_i, w_j) \in q^{k-1} \setminus m^{k-1} \times n^{k-1}$ it holds that $l(u_i, w_j) = \emptyset$ or $\{\perp\}$ and $(u_i, w_j)$ does not have any $r$-successor and $\forall r$-successor. Furthermore, $l_{G\times H}(u, w) = \emptyset$ and $(u, w)$ does not have any $r$-successor and $\forall r$-successor for all $(u, w) \in V_{G\times H}$ such that $((u, v)\epsilon(u_i, w_j)) \in E^\epsilon_{G\times H}$, $(v, w)$ is a $\forall r$-successor and $(u_i, w_j) \in q^{k-1}$.
   (b) $\mathsf{label}(m^{k-1}, n^{k-1}) = \mathsf{label}(q^{k-1}) \neq \{\perp\}$; and for all $(v_i, z_j) \in q^{k-1} \setminus m^{k-1} \times n^{k-1}$ it holds that $l(v_i, z_j) = \emptyset$ and $(v_i, z_j)$ does not have any $r$-successor and $\forall r$-successor. Furthermore, if $q^{k-1} = V^\epsilon$ then $l_{G\times H}(u, w) = \emptyset$ and $(u, w)$ does not have any $r$-successor and $\forall r$-successor for all $(u, w) \in V_{G\times H}$ such that $((u, v)\epsilon(u_i, w_j)) \in E^\epsilon_{G\times H}$, $(v, w)$ is a $\forall r$-successor and $(u_i, w_j) \in q^{k-1}$.

Note that this hypothesis is verified if $\mathcal{G}^\epsilon$ and $\mathcal{H}^\epsilon$ are normalization graphs.

I) $\mathsf{label}(m^{k-1}) = \{\perp\}$ and $\mathsf{label}(n^{k-1}) \neq \{\perp\}$ (or $\mathsf{label}(m^{k-1}) \neq \{\perp\}$ and $\mathsf{label}(n^{k-1}) = \{\perp\}$).

By the induction hypothesis, we have that $l_G(u_i) = \{\perp\}$ for some $u_i \in m^{k-1}$ and, $l_G(u_{i'}) = \emptyset$, $u_{i'}$ does not have any $\forall r$-successor and $r$-successor for all $u_{i'} \in m^{k-1}$, $u_{i'} \neq u_i$. Furthermore, from $\mathsf{label}(n^{k-1}) \neq \{\perp\}$ the definition of function $\mathsf{label}$ yields that $l_H(w_j) \neq \{\perp\}$ for all $w_j \in n^{k-1}$. From Definition 9 (product), it is that the product graph $\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon$ contains the subgraph $(\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon)((u_i, w_1), ..., (u_i, w_n))$ which is obtained from the subgraph $\mathcal{H}^\epsilon(w_1, ..., w_n)$ where $l_G(u_i) = \{\perp\}$. By the induction hypothesis, we have $q^{k-1} = V^\epsilon$, $\{(u_i, w_1), ..., (u_i, w_n)\} \subseteq q^{k-1}$ such that $l_G(u_i) = \{\perp\}$, and $l_{G\times H}(u_{i'}, w_j) = \emptyset$, $(u_{i'}, w_j)$ does not have any $r$-successor and $\forall r$-successor for all $(v_{i'}, z_j) \in q^{k-1} \setminus \{(u_i, w_1), ..., (u_i, w_n)\}$.

On the other hand, according to the definition of product in [3] we obtain that the subtree $\mathbf{B}(\mathcal{G}^\epsilon)(m^{k-1}) \times \mathbf{B}(\mathcal{H}^\epsilon)(n^{k-1})$ is equal to the subtree $\mathbf{B}(\mathcal{H}^\epsilon)(n^{k-1})$. This implies that $\mathbf{B}((\mathcal{G}^\epsilon \times \mathcal{H}^\epsilon)(q^{k-1})) = \mathbf{B}(\mathcal{G}^\epsilon)\ (m^{k-1}) \times \mathbf{B}(\mathcal{H}^\epsilon)(n^{k-1})$.

II) $\mathsf{label}(m^{k-1}) \neq \{\perp\}$ and $\mathsf{label}(n^{k-1}) \neq \{\perp\}$.

1. Let $m^k = (v_1, ..., v_a)$ and $n^k = (z_1, ..., z_b)$ be $\forall r$-neighbourhoods respectively of $m^{k-1}$ and $n^{k-1}$. Let $q^k$ be the $\forall r$-neighbourhood of $q^{k-1}$. First, we show that $V^\forall_{G\times H}(q^{k-1}) \neq \emptyset$ iff $V^\forall_G(m^{k-1}) \neq \emptyset$ and $V^\forall_H(n^{k-1}) \neq \emptyset$. Assume that $v_i \in V^\forall_G(m^{k-1})$ and $z_j \in V^\forall_H(n^{k-1})$ i.e $(u_h \forall r v_i) \in E_G$ and $(w_l \forall r z_j) \in E_H$ where $u_h \in m^{k-1}$ and $u_l \in n^{k-1}$. By the induction hypothesis, we have $\{(u_1, w_1), ..., (u_m, w_n)\} \subseteq q^{k-1}$. By Definition 9 (product), that means that $(v_i, z_j) \in V^\forall_{G\times H}(q^{k-1})$. Conversely, assume that $(v_i, z_j) \in V^\forall_{G\times H}(q^{k-1})$ i.e $((u_h, u_l)\forall r(v_i, z_j)) \in E_{G\times H}$ where $(u_h, w_l) \in q^{k-1}$. By the induction hypothesis, we have $(u_h, w_l) \in \{(u_1, w_1), ..., (u_m, w_n)\}$ since $\{(u_1, w_1), ..., (u_m, w_n)\} \subseteq q^{k-1}$ and $(u_{h'}, w_{l'})$ does not have any $\forall r$-successor for all $(u_{h'}, w_{l'}) \in q^{k-1} \setminus \{(u_1, w_1), ..., (u_m, w_n)\}$. By Definition 9 (product), that means that $v_i \in V^\forall_G(m^{k-1})$ and $z_j \in V^\forall_H(n^{k-1})$.

By consequent, we obtain there exists a $\forall r$-neighbourhood of $q^{k-1}$ iff there exist $\forall r$-neighbourhoods of $m^{k-1}$ and $n^{k-1}$.

1.1) Assume that $\mathsf{label}(m^k) = \{\perp\}$ and $\mathsf{label}(n^k) = \{\perp\}$.

The induction hypothesis yields that $l_G(v_0) = \{\perp\}$ for some $v_0 \in m^k$ and $l_H(z_0) = \{\perp\}$ for some $z_0 \in$

$n^k$ and $m^k = V^{\epsilon}_G(m^{k-1})$ and $n^k = V^{\epsilon}_H(n^{k-1})$ are the $\forall r$-neighbourhoods of $m^{k-1}$ and $n^{k-1}$ respectively.
We show that $q^k = V^{\epsilon}_{GH}(q^{k-1})$,
$V^{\epsilon}_G(m^{k-1}) \times V^{\epsilon}_H(n^{k-1}) \cup V^{\forall}_{GE} \cup V^{\forall}_{HE} = q^k$,
$l_{G\times H}(v_i, z_j) = \{\perp\}$ for some $(v_i, z_j) \in q^k$;
$l_{G\times H}(v_{i'}, z_{j'}) = \emptyset$, $(v_{i'}, z_{j'})$ does not have any $r$-successor and $\forall r$-successor for all $(v_{i'}, z_{j'}) \in q^k$; and$l_{G\times H}(v_l, z_h) = \emptyset$, $(v_l, z_h)$ does not have any $r$-successor and $\forall r$-successor for all $(v_l, z_h)$ such that $(v_l, z_h) \in V^{\forall}(q^{k-1})$, $(v_l, z_h)\epsilon(v_i, z_j) \in E^{\epsilon}_{G\times H}$, $(v_i, z_j) \in q^k$ (*) . Note that
$V^{\forall}_{GE} := \{(v, z) \mid v\in V^{\forall}_G(m^{k-1}),\ (v\epsilon v') \in E^{\epsilon}_G,\ v' \in V^{\epsilon}_G(m^{k-1}),\ z \in V^{\epsilon}_H(n^{k-1})\}$ and
$V^{\forall}_{HE} := \{(v, z) \mid v \in V^{\epsilon}_G(m^{k-1}),\ z \in V^{\forall}_H(n^{k-1})\ ,\ (z\epsilon z') \in E^{\epsilon}_H,\ z' \in V^{\epsilon}_H(n^{k-1})\}$.
Assume that $v_i \in V^{\epsilon}_G(m^{k-1})$ and $z_j \in V^{\epsilon}_H(n^{k-1})$. This means that $l_G(v_i) = \{\perp\}$ or $\emptyset$, $p(v_i) \in m^{k-1}$,$l_G(v_l) = \emptyset$, $(v_l\epsilon v_i) \in E^{\epsilon}_G$, $v_l \in V^{\forall}(m^{k-1})$, $v_i \notin V^{\forall}(m^{k-1})$, $v_i$ does not have any $\forall r$-successor and $r$-successor, and $l_H(z_j) = \{\perp\}$ or $\emptyset$, $p(z_j) \in n^{k-1}$, $l_H(z_h) = \emptyset$, $(z_h\epsilon z_j) \in E^{\epsilon}_H$, $z_h \in V^{\forall}(n^{k-1})$, $z_j \notin V^{\forall}(n^{k-1})$, $z_j$ does not have any $\forall r$-successor and $r$-successor. Hence, we have $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $l_{G\times H}(v_l, z_h) = \emptyset$, $((v_l, z_h)\epsilon(v_i, z_j)) \in E^{\epsilon}_{G\times H}$, $p(v_i, z_j) = (p(v_i), p(z_j)) \in q^{k-1}$, $(v_i, z_j) \notin V^{\forall}(q^{k-1})$,
$l_{G\times H}(v_i, z_j) = \{\perp\}$ or $\emptyset$. Therefore, according to Definition 6 (neighbourhood), $(v_i, z_j) \in V^{\epsilon}_{GH}(q^{k-1})$. Thus, $V^{\epsilon}_G(m^{k-1}) \times V^{\epsilon}_H(n^{k-1}) \subseteq q^k$ and $q^k = V^{\epsilon}_{GH}(q^{k-1})$.

Assume that $(v_i, z_j) \in V^{\forall}_{GE}$. By the definition, we have $((v_i, z_h)\epsilon(v_i, z_j)) \in E^{\epsilon}_{G\times H}$, $(v_i, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $(v_i, z_j) \notin V^{\forall}_{GH}(q^{k-1})$, $p(v_i, z_j) \in q^{k-1}$ where $z_h \in V^{\forall}_H(n^{k-1})$, $(z_h\epsilon z_j) \in E^{\epsilon}_H$, $z_j \in V^{\epsilon}_H(n^{k-1})$. This implies that $(v_i, z_j) \in V^{\epsilon}_{GH}(q^{k-1})$ and $l_{G\times H}(v_i, z_h) = \emptyset$, $l_{G\times H}(v_i, z_j) = \emptyset$ and $(v_i, z_j)$ does not have any successor. Similarly, if $(v_i, z_j) \in V^{\forall}_{HE}$ then $(v_i, z_j) \in V^{\epsilon}_{GH}(q^{k-1})$ and $l_{G\times H}(v_l, z_j) = \emptyset$, $l_{G\times H}(v_i, z_j) = \emptyset$ and $(v_i, z_j)$ does not have any successor where $v_l \in V^{\forall}_G(m^{k-1})$, $(v_l\epsilon v_i) \in E^{\epsilon}_G$, $v_i \in V^{\epsilon}_G(m^{k-1})$.

Conversely, assume that $(v_i, z_j) \in q^k = V^{\epsilon}_{GH}(q^{k-1})$. According to Definition 6 (neighbourhood), we have that $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_i, z_j)) \in E^{\epsilon}_{G\times H}$, $(v_i, z_j) \notin V^{\forall}_{GH}(q^{k-1})$, $p(v_i, z_j) \in q^{k-1}$. From Definition 9 (product), we have $v_l \in V^{\forall}(m^{k-1})$, $(v_l\epsilon v_i) \in E^{\epsilon}_G$ and $z_h \in V^{\forall}(n^{k-1})$, $(z_h\epsilon z_j) \in E^{\epsilon}_G$, $p(v_i) \in m^{k-1}$, $p(z_j) \in z^{k-1}$, and either $v_i \in V^{\forall}(m^{k-1})$, $z_j \in V^{\epsilon}_H(n^{k-1})$ or $v_i \in V^{\epsilon}_G(m^{k-1})$, $z_j \in V^{\forall}(n^{k-1})$. This implies that $(v_i, z_j) \in V^{\epsilon}_G(m^{k-1}) \times V^{\epsilon}_H(n^{k-1}) \cup V^{\forall}_{GE} \cup V^{\forall}_{HE}$. It is obvious that for all $(u_i, v_j) \in V^{\epsilon}_G(m^{k-1}) \times V^{\epsilon}_H(n^{k-1}) \cup V^{\forall}_{GE} \cup V^{\forall}_{HE}$ it holds that $l_{G\times H}(v_i, z_j) = \{\perp\}$ or $\emptyset$ and $(v_i, z_j)$ does not have any successor. Furthermore, $l_{G\times H}(v, z) = \{\perp\}$ or $\emptyset$, $(v, z)$ does not have any successor for all $(v, z)$ such that $(v, z) \in V^{\forall}(q^{k-1})$, $(v, z)\epsilon(v_i, z_j) \in E^{\epsilon}_{G\times H}$, $(v_i, z_j) \in q^k$.
1.2) Assume that $\mathsf{label}(m^k) = \{\perp\}$ and $\mathsf{label}(n^k) \neq \{\perp\}$ (or $\mathsf{label}(m^k) \neq \{\perp\}$ and $\mathsf{label}(n^k) = \{\perp\}$). The induction hypothesis yields that $m^k = V^{\epsilon}_G(m^{k-1})$ and $m^k$ is the $\forall r$-neighbourhoods of $m^{k-1}$. This means that for all $v_i \in m^k$ we have $l_G(v_i) = \{\perp\}$ or $\emptyset$, $p(v_i) \in m^{k-1}$,$l_G(v_l) = \emptyset$, $(v_l\epsilon v_i) \in E^{\epsilon}_G$, $v_l \in V^{\forall}_G(m^{k-1})$, $v_i \notin V^{\forall}_G(m^{k-1})$, $v_i$ does not any $\forall r$-successor and $r$-successor; $l_G(v_l) = \emptyset$, $v_l$ does not have any $\forall r$-successor and $r$-successor for all $v_l$ such that $v_l \in V^{\forall}_G(m^{k-1})$, $(v_l\epsilon v_i) \in E^{\epsilon}_G$, $v_i \in V^{\epsilon}_G(m^{k-1})$. Furthermore, $l_H(z) \neq \{\perp\}$ for all $z \in n^k$. These imply that $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_i, z_h)) \in E^{\epsilon}_{G\times H}$, $(v_i, z_h) \notin V^{\forall}_{GH}(q^{k-1})$, $p(v_i, z_h) \in q^{k-1}$ where $z_h \in V^{\forall}_H(n^{k-1})$. Thus, by Definition 6 (neighbourhood), $(v_i, z_h) \in V^{\epsilon}_{GH}(q^{k-1}) \neq \emptyset$ and $q^k = V^{\epsilon}_{GH}(q^{k-1})$.
Let $v_0 \in m^k$ such that $l_G(v_0) = \{\perp\}$. According to the definition of product in [3] we obtain that the subtree $\mathbf{B}(\mathcal{G}^{\epsilon})(m^k) \times \mathbf{B}(\mathcal{H}^{\epsilon})(n^k)$ is equal to the subtree $\mathbf{B}(\mathcal{H}^{\epsilon})(n^k)$. This implies that $(m^k, n^k) = n^k$. On the other hand, from Definition 9 (product), we have that the product graph $\mathcal{G}^{\epsilon} \times \mathcal{H}^{\epsilon}$ contains the subgraph $(\mathcal{G}^{\epsilon} \times \mathcal{H}^{\epsilon})((v_0, z_1), ..., (v_0, z_b))$ where this subgraph is obtained from the subgraph $\mathcal{H}^{\epsilon}(z_1, ..., z_b)$. We have to prove that $\{(v_0, z_1), ..., (v_0, z_b)\} \subseteq q^k$ and $l_{G\times H}(v_{i'}, z_{j'}) = \emptyset$, $(v_{i'}, z_{j'})$ does not have any $\forall r$-successor or $r$-successor for all $(v_{i'}, z_{j'}) \in q^k \setminus \{(v_0, z_1), ..., (v_0, z_b)\}$. Furthermore, we show that $l_{G\times H}(v_l, z_h) = \emptyset$, $(v_l, z_h)$ does not have any $\forall r$-successor and $r$-successor for all $(v_l, z_h)$ such that $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_i, z_j))$ and $(v_i, z_j) \in q^k$.
Assume that $n^k = V^{\forall}_H(n^{k-1})$. Similar to above, we have $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_0, z_h)) \in E^{\epsilon}_{G\times H}$, $(v_0, z_h) \notin V^{\forall}_{GH}(q^{k-1})$, $p(v_0, z_h) \in q^{k-1}$ for all $z_h \in V^{\forall}_H(n^{k-1})$ where $v_l \in V^{\forall}_G(m^{k-1})$, $l_G(v_0) = \{\perp\}$ such that $(v_l\epsilon v_0) \in E^{\epsilon}_G$, $v_0 \notin V^{\forall}_G(m^{k-1})$, $p(v_0) \in m^{k-1}$. Therefore, $\{(v_0, z_1), ..., (v_0, z_b)\} \subseteq q^k$. Let now $(v_{i'}, z_{j'}) \in q^k \setminus \{(v_0, z_1), ..., (v_0, z_b)\}$. By Definition 6 (neighbourhood), we have $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_{i'}, z_{j'})) \in E^{\epsilon}_{G\times H}$, $(v_{i'}, z_{j'}) \notin V^{\forall}_{GH}(q^{k-1})$, $p(v_{i'}, z_{j'}) \in q^{k-1}$. This implies that $v_{i'} \in V^{\epsilon}_G(m^{k-1})$ or if $v_{i'}$is a $\forall r$-successor then $l_G(v_{i'}) = \emptyset$ and $v_{i'}$ does not have any $\forall r$-successor and $r$-

successor (the property (*) of $V^{\epsilon}$is proven above). Therefore, $l_{G\times H}(v_{i'}, z_{j'}) = \emptyset$ and $(v_{i'}, z_{j'})$ does not have any $\forall r$-successor and $r$-successor. Furthermore, if $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_i, z_j))$ and $(v_i, z_j) \in q^k$ then either $v_l \in V^{\epsilon}_G(m^{k-1})$ or $v_l \in V^{\forall}_G(m^{k-1})$, $(v_l\epsilon v) \in E^{\epsilon}_G$, $v \in V^{\epsilon}_G(m^{k-1})$. This implies that $l_{G\times H}(v_l, z_h) = \emptyset$, $(v_l, z_h)$ does not have any $\forall r$-successor and $r$-successor.

Assume that $n^k = V^{\epsilon}_H(n^{k-1})$. We have $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_0, z_j)) \in E^{\epsilon}_{G\times H}$, $(v_0, z_j) \notin V^{\forall}_{GH}(q^{k-1})$, $p(v_0, z_j) \in q^{k-1}$ for all $z_j \in V^{\epsilon}_H(n^{k-1})$ where $v_l \in V^{\forall}_G(m^{k-1})$, $l_G(v_0) = \{\perp\}$ such that $(v_l\epsilon v_0) \in E^{\epsilon}_G$, $v_0 \notin V^{\forall}_G(m^{k-1})$, $p(v_0) \in m^{k-1}$. This implies that $\{(v_0, z_1), ..., (v_0, z_b)\} \subseteq q^k$. Let now $(v_{i'}, z_{j'}) \in q^k \setminus \{(v_0, z_1), ..., (v_0, z_b)\}$. Similar to above, we have that $v_{i'} \in V^{\epsilon}_G(m^{k-1})$ or if $v_{i'}$is a $\forall r$-successor then $l_G(v_{i'}) = \emptyset$ and $v_{i'}$ does not have any successor (the property (*) of $V^{\epsilon}$is proven above). Therefore, $l_{G\times H}(v_{i'}, z_{j'}) = \emptyset$ and $(v_{i'}, z_{j'})$ does not have any $\forall r$-successor and $r$-successor. Furthermore, if $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_i, z_j))$ and $(v_i, z_j) \in q^k$ then we can show that $l_{G\times H}(v_l, z_h) = \emptyset$, $(v_l, z_h)$ does not have any $\forall r$-successor and $r$-successor.

Thus, we have shown $\{(v_0, z_1), ..., (v_0, z_b)\} \subseteq q^k$ and $l_{G\times H}(v_{i'}, z_{j'}) = \emptyset$, $(v_{i'}, z_{j'})$ does not have any $\forall r$-successor and $r$-successor for all $(v_{i'}, z_{j'}) \in q^k \setminus \{(v_0, z_1), ..., (v_0, z_b)\}$. By consequent, we obtain that $\mathsf{label}(q^k) = \mathsf{label}(m^k) \cap \mathsf{label}(n^k)$ and $\mathbf{B}((\mathcal{G}^{\epsilon} \times \mathcal{H}^{\epsilon})(q^k)) = \mathbf{B}(\mathcal{G}^{\epsilon})\ (m^k) \times \mathbf{B}(\mathcal{H}^{\epsilon})(n^k)$.

1.3) Assume that $\mathsf{label}(m^k) \neq \{\perp\}$ and $\mathsf{label}(n^k) \neq \{\perp\}$. We consider the three following cases:

i) Assume that $m^k = V^{\forall}_G(m^{k-1})$ and $n^k = V^{\forall}_H(n^{k-1})$.

It holds that $q^k = V^{\forall}_{GH}(q^{k-1})$ since if $q^k = V^{\epsilon}_{GH}(q^{k-1})$ then $m^k = V^{\epsilon}_G(m^{k-1})$ or $n^k = V^{\epsilon}_H(n^{k-1})$. Moreover, we have

$V^{\forall}_{GH}(q^{k-1}) = \{(v_1, z_1), ..., (v_a, z_b)\}$. Thus, Lemma A yields that $\mathsf{label}(q^k) = \mathsf{label}(m^k) \cap \mathsf{label}(n^k)$.

ii) Assume that $m^k = V^{\epsilon}_G(m^{k-1})$ and $n^k = V^{\forall}_H(n^{k-1})$ (or $m^k = V^{\forall}_G(m^{k-1})$, $n^k = V^{\epsilon}_H(n^{k-1})$). We show that

$q^k = V^{\epsilon}_{GH}(q^{k-1}) = \{(v_1, z_1), ..., (v_a, z_b)\}$.

Let $(v_i, z_h) \in \{(v_1, z_1), ..., (v_a, z_b)\}$. We have that $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_i, z_h)) \in E^{\epsilon}_{G\times H}$, $(v_i, z_h) \notin V^{\forall}_{GH}(q^{k-1})$, $p(v_i, z_h) \in q^{k-1}$ for all $z_h \in V^{\forall}_H(n^{k-1})$ and for all $v_i \in V^{\epsilon}_G(m^{k-1})$ such that $v_l \in V^{\forall}_G(m^{k-1})$, $(v_l\epsilon v_i) \in E^{\epsilon}_G$, $v_i \notin V^{\forall}_G(m^{k-1})$, $p(v_i) \in m^{k-1}$. Therefore, $(v_i, v_h) \in V^{\epsilon}_{GH}(q^{k-1})$ and $\{(v_1, z_1), ..., (v_b, z_b)\} \subseteq q^k$. Let now $(v_{i'}, z_{j'}) \in q^k \setminus \{(v, z_1), ..., (v_0, z_b)\}$. Conversely, let $(v_i, z_h) \in V^{\epsilon}_{GH}(q^{k-1})$. We have $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_i, z_j)) \in E^{\epsilon}_{G\times H}$, $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $(v_i, z_j) \notin V^{\forall}_{GH}(q^{k-1})$, $p(v_i, z_j) \in q^{k-1}$. This implies that $v_l \in V^{\forall}_G(m^{k-1})$, $z_h \in V^{\forall}_H(n^{k-1})$, and $v_i \notin V^{\forall}_G(m^{k-1})$, $p(v_i) \in m^{k-1}$ or $z_j \notin V^{\forall}_H(n^{k-1})$, $p(z_j) \in n^{k-1}$. If $z_j \notin V^{\forall}_H(n^{k-1})$, $p(z_j) \in n^{k-1}$ then $z_j \in V^{\epsilon}_H(n^{k-1})$ and thus $n^k = V^{\epsilon}_H(n^{k-1})$. This is a contradiction. Therefore, $V^{\epsilon}_{GH}(q^{k-1}) = \{(v_1, z_1), ..., (v_a, z_b)\}$. As above, we can show that if $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_i, z_j))$ and $(v_i, z_j) \in q^k$ then $l_{G\times H}(v_l, z_h) = \emptyset$, $(v_l, z_h)$ does not have any $\forall r$-successor and $r$-successor. In this case, it is obvious that, by Lemma A, $\mathsf{label}(q^k) = \mathsf{label}(m^k) \cap \mathsf{label}(n^k)$.

iii) Assume that $m^k = V^{\epsilon}_G(m^{k-1})$ and $n^k = V^{\epsilon}_H(n^{k-1})$.

We show that $q^k = V^{\epsilon}_{GH}(q^{k-1})$ and

$\{(v_1, z_1), ..., (v_a, z_b)\} \subseteq q^k$ and $l_{G\times H}(v_{i'}, z_{j'}) = \emptyset$, $(v_{i'}, z_{j'})$ does not have any $\forall r$-successor and $r$-successor for all $(v_{i'}, z_{j'}) \in q^k \setminus \{(v_1, z_1), ..., (v_a, z_b)\}$. Furthermore, we show that $l_{G\times H}(v_l, z_h) = \emptyset$, $(v_l, z_h)$ does not have any $\forall r$-successor and $r$-successor for all $(v_l, z_h)$ such that $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_i, z_j))$ and $(v_i, z_j) \in q^k$.

We have $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_i, z_j)) \in E^{\epsilon}_{G\times H}$, $(v_i, z_j) \notin V^{\forall}_{GH}(q^{k-1})$, $p(v_i, z_j) \in q^{k-1}$ for all $v_i \in V^{\epsilon}_G(m^{k-1})$ and $z_j \in V^{\epsilon}_H(n^{k-1})$. This implies that $\{(v_0, z_1), ..., (v_0, z_b)\} \subseteq q^k$. Let now $(v_{i'}, z_{j'}) \in q^k \setminus \{(v_1, z_1), ..., (v_1, z_b)\}$. Similar to above, we have that $v_{i'} \in V^{\epsilon}_G(m^{k-1})$ or if $v_{i'}$is a $\forall r$-successor then $l_G(v_{i'}) = \emptyset$ and $v_{i'}$ does not have any $\forall r$-successor and $r$-successor (the property (*) of $V^{\epsilon}$is proven above). Therefore, $l_{G\times H}(v_{i'}, z_{j'}) = \emptyset$ and $(v_{i'}, z_{j'})$ does not have any $\forall r$-successor and $r$-successor. Furthermore, if $(v_l, z_h) \in V^{\forall}_{GH}(q^{k-1})$, $((v_l, z_h)\epsilon(v_i, z_j)) \in E^{\epsilon}_{G\times H}$ and $(v_i, z_j) \in q^k$ then $v_l \in V^{\forall}_G(m^{k-1})$, $z_h \in V^{\forall}_G(m^{k-1})$ such that $(v_l\epsilon v_i) \in E^{\epsilon}_G$, $v_i \in V^{\epsilon}_G(m^{k-1})$ or $(z_h\epsilon z_j) \in E^{\epsilon}_H$, $z_j \in V^{\epsilon}_H(n^{k-1})$. This implies that $l_{G\times H}(v_l, z_h) = \emptyset$, $(v_l, z_h)$ does not have any $\forall r$-successor and $r$-successor. In this case, it is obvious that, by Lemma A, $\mathsf{label}(q^k) = \mathsf{label}(m^k) \cap \mathsf{label}(n^k)$.

To sum up, the isomorphism is extended as follows: $\phi(m^k, n^k) := q^k$ where $m^k$, $n^k$and $q^k$ are $\forall r$-neighbourhoods respectively of $m^{k-1}$, $n^{k-1}$and $q^{k-1}$.

2. Let $m^k_i$ and $n^k_j$ be $r$-neighbourhoods respectively of $m^{k-1}$ and $n^{k-1}$. Let $q^k_{ij}$ be a $r$-neighbourhood of $q^{k-1}$ such that

$m^k_i = \{v_i\} \cup V^{\epsilon}_{GE}$ , $V^{\epsilon}_{GE} = \{v'_l | \ (v_i\epsilon v'_l) \in E^{\epsilon}_G, p(v'_l) \in m^{k-1}\}$, $(u_i r v_i) \in E_G$,

$n_j^k = \{z_j\} \cup V_{HE}^{\epsilon}$ , $V_{HE}^{\epsilon} = \{z_l' | \ (z_j \epsilon z_l') \in E_H^{\epsilon}, p(z_l') \in n^{k-1}\}$, $(w_j r z_j) \in E_H$,
$q_{ij}^k = \{(v_i, z_j)\} \cup \bigcup_{(v_l, y_{l'}) \in V^{\epsilon}} (v_l, y_{l'})$, where
$V^{\epsilon} = \{(v_l, z_{l'}) | \ (v_i, z_j)\epsilon(v_l, z_{l'}), p(v_l, z_{l'}) \in q^{k-1}\}$
First, we show that there exists $(v_i, z_j) \in q_{ij}^k$ iff $v_i \in m_i^k$ and $z_j \in n_i^k$. Assume that $v_i \in m_i^k$ and $z_j \in n_j^k$ i.e $(u_h r v_i) \in E_G$ and $(w_l r z_j) \in E_H$ where $u_h \in m^{k-1}$ and $u_l \in n^{k-1}$. By the induction hypothesis, we have $\{(u_1, w_1), ..., (u_m, w_n)\} \subseteq q^{k-1}$. By Definition 9 (product), that means that $(v_i, z_j) \in q_{ij}^k$. Conversely, assume that $(v_i, z_j) \in q_{ij}^k$ i.e $((u_h, u_l) r (v_i, z_j)) \in E_{G \times H}$ where $(u_h, w_l) \in q^{k-1}$. By the induction hypothesis, we have $(u_h, w_l) \in \{(u_1, w_1), ..., (u_m, w_n)\}$ since $\{(u_1, w_1), ..., (u_m, w_n)\} \subseteq q^{k-1}$ and $(u_{h'}, w_{l'})$ does not have any $r$-successor for all $(u_{h'}, w_{l'}) \in q^{k-1} \setminus \{(u_1, w_1), ..., (u_m, w_n)\}$. By Definition 9 (product), that means that $v_i \in m_i^k$ and $z_j \in n_j^k$.
We have to prove that $m_i^k \times n_j^k = q_{ij}^k$ and $\mathsf{label}(q_{ij}^k) = \mathsf{label}(m_i^k) \cap \mathsf{label}(n_j^k)$. Hence, we have that $\mathsf{label}(q_{ij}^k) \neq \{\bot\}$ if
$\mathsf{label}(m^k) \neq \{\bot\}$ and $\mathsf{label}(n^k) \neq \{\bot\}$ (if $\mathcal{G}^{\epsilon}$ and $\mathcal{H}^{\epsilon}$ are normalization graphs then $\mathsf{label}(m^k) \neq \{\bot\}$ and $\mathsf{label}(n^k) \neq \{\bot\}$ ). From Lemma A, it is that $\mathsf{label}(q_{ij}^k) = \mathsf{label}(m_i^k) \cap \mathsf{label}(n_j^k)$.
What remains to be shown is that i) for each $(v_l, z_{l'}) \in V^{\epsilon}$ we obtain that $v_l \in V_{GE}^{\epsilon}$ and $z_{l'} \in V_{HE}^{\epsilon}$ ii) for each $v_l \in V_{GE}^{\epsilon}$ and $z_{l'} \in V_{HE}^{\epsilon}$ we obtain that $(v_l, z_{l'}) \in V^{\epsilon}$.

1. for each $(v_l, y_{l'}) \in V^{\epsilon}$ we obtain that $v_l \in V_{GE}^{\epsilon}$ and $y_{l'} \in V_{HE}^{\epsilon}$.
   Let $(v_l, y_{l'}) \in V^{\epsilon}$. This yields that there exist edges $(v_i, y_j)\epsilon(v_l, y_{l'}) \in E_{G \times H}^{\epsilon}$, $(u_i, w_j) r (v_i, y_j) \in E_{G \times H}$ and $p(v_l, y_{l'}) = (p(v_l), p(y_{l'})) \in q^{k-1}$. This implies that $(v_i \epsilon v_l) \in E_G^{\epsilon}$, $(y_j \epsilon y_{l'}) \in E_H^{\epsilon}$ where $p(v_l) \in m^{k-1}$, $p(y_{l'}) \in n^{k-1}$. From $u_i \in m^{k-1}$, $(u_i r v_i) \in E_G$, $p(v_l) \in m^{k-1}$ and $(v_i \epsilon v_l) \in E_G^{\epsilon}$, we obtain $v_l \in V_{GE}^{\epsilon}$. From $w_j \in n^{k-1}$, $(w_j r y_j) \in E_H$, $p(y_{l'}) \in n^{k-1}$ and $(y_j \epsilon y_{l'}) \in E_H^{\epsilon}$, we obtain $y_{l'} \in V_{HE}^{\epsilon}$.
2. for each $v_l \in V_{GE}^{\epsilon}$ and $y_{l'} \in V_{HE}^{\epsilon}$ we obtain that $(v_l, y_{l'}) \in V^{\epsilon}$.
   Let $v_l \in V_{GE}^{\epsilon}$ and $y_{l'} \in V_{HE}^{\epsilon}$. This yields that there are $\epsilon$-edges $(v_i \epsilon v_l) \in E_G^{\epsilon}$, $(y_j \epsilon y_{l'}) \in E_H^{\epsilon}$, $p(v_l) \in m^{k-1}$, $p(y_{l'}) \in n^{k-1}$ and $(u_i r v_i) \in E_G$, $(w_j r y_j) \in E_H$. This implies that $(v_i, y_j)\epsilon(v_l, y_{l'}) \in E_{G \times H}^{\epsilon}$, $((u_i, w_j) r (v_i, y_j)) \in E_{G \times H}$ and $p(v_l, y_{l'}) = (p(v_l), p(y_{l'})) \in x^{k-1}$. Thus, $(v_l, y_{l'}) \in V^{\epsilon}$.

The isomorphism is extended as follows: $\phi(m_i^k, n_j^k) := q_{ij}^k$ where $m_i^k$, $n_j^k$and $q_{ij}^k$ are $r$-neighbourhoods respectively of $m^{k-1}$, $n^{k-1}$and $q^{k-1}$. The induction principle guarantees that $\phi$ is an isomorphism between trees $\mathbf{B}(\mathcal{G}^{\epsilon} \times \mathcal{H}^{\epsilon})$ and $\mathbf{B}(\mathcal{G}^{\epsilon}) \times \mathbf{B}(\mathcal{H}^{\epsilon})$. ■

**Proposition 4** *Let $C = C_1 \sqcup ... \sqcup C_n$ be an $\mathcal{ALC}$-concept description where $\bot \sqsubset C_1, ..., C_n$. The approximation of $C$ by $\mathcal{ALE}$-concept description can be computed as follows:*

$$approx_{\mathcal{ALE}}(C) \equiv \text{lcs}\{approx_{\mathcal{ALE}}(C_1), \ ... \ , approx_{\mathcal{ALE}}(C_n)\}$$

*Proof.* The proof is direct from the definitions of *lcs* and *approx*.
First, prove the proposition with $n = 2$. We have that
$C_1 \sqcup C_2 \sqsubseteq lcs\{approx_{\mathcal{ALE}}(C_1), \ approx_{\mathcal{ALE}}(C_2)\}$ since $C_1 \sqsubseteq approx_{\mathcal{ALE}}(C_1)$, $C_2 \sqsubseteq approx_{\mathcal{ALE}}(C_2)$, $approx_{\mathcal{ALE}}(C_1) \sqsubseteq lcs\{approx_{\mathcal{ALE}}(C_1), approx_{\mathcal{ALE}}(C_2)\}$ and $approx_{\mathcal{ALE}}(C_2) \sqsubseteq lcs\{approx_{\mathcal{ALE}}(C_1), approx_{\mathcal{ALE}}(C_2)\}$.
Assume that there exists an $\mathcal{ALE}$-concept description $D$ such that
$C_1 \sqcup C_2 \sqsubseteq D \sqsubseteq lcs\{approx_{\mathcal{ALE}}(C_1), approx_{\mathcal{ALE}}(C_2)\}$ (**).
We show that $approx_{\mathcal{ALE}}(C_1) \sqcup approx_{\mathcal{ALE}}(C_2) \not\sqsubseteq D$ is impossible.
Indeed, there exist an interpretation $(\Delta, .^{\mathcal{I}})$ and an individual $d^{\mathcal{I}} \in \Delta$ such that $d^{\mathcal{I}} \in (approx_{\mathcal{ALE}}(C_1) \sqcup approx_{\mathcal{ALE}}(C_2))^{\mathcal{I}}$ $(\bot \sqsubset C_1, C_2)$ and $d^{\mathcal{I}} \notin D^{\mathcal{I}}$.
There are the two following possibilities:

– If $d^{\mathcal{I}} \in (approx_{\mathcal{ALE}}(C_1))^{\mathcal{I}}$ and $d^{\mathcal{I}} \notin D^{\mathcal{I}}$, then $C_1 \sqsubseteq D \sqcap approx_{\mathcal{ALE}}(C_1) \sqsubset approx_{\mathcal{ALE}}(C_1)$, which contradicts the approximation definition since $D \sqcap approx_{\mathcal{ALE}}(C_1)$ is an $\mathcal{ALE}$-concept description.
– If $d^{\mathcal{I}} \in (approx_{\mathcal{ALE}}(C_2))^{\mathcal{I}}$ and $d^{\mathcal{I}} \notin D^{\mathcal{I}}$, then $C_2 \sqsubseteq D \sqcap approx_{\mathcal{ALE}}(C_2) \sqsubset approx_{\mathcal{ALE}}(C_{12})$, which contradicts the approximation definition since $D \sqcap approx_{\mathcal{ALE}}(C_2)$ is an $\mathcal{ALE}$-concept description.

Hence, we obtain that
$approx_{\mathcal{ALE}}(C_1) \sqcup approx_{\mathcal{ALE}}(C_2) \sqsubseteq D$. This implies that
$lcs(approx_{\mathcal{ALE}}(C_1), \ approx_{\mathcal{ALE}}(C_2)\} \equiv D$ since the hypothesis (**),

$approx_{\mathcal{ALE}}(C_1) \sqsubseteq D$, $approx_{\mathcal{ALE}}(C_2) \sqsubseteq D$ and $lcs\{approx_{\mathcal{ALE}}(C_1), approx_{\mathcal{ALE}}(C_2)\}$ is the least $\mathcal{ALE}$-concept description such that

$$approx_{\mathcal{ALE}}(C_1) \sqsubseteq lcs\{approx_{\mathcal{ALE}}(C_1), approx_{\mathcal{ALE}}(C_2)\},$$

$$approx_{\mathcal{ALE}}(C_2) \sqsubseteq lcs\{approx_{\mathcal{ALE}}(C_1), approx_{\mathcal{ALE}}(C_2)\}.$$

By induction on $n$, the proposition can be proven for $n > 2$ by using the following property of the $lcs$:

$$lcs\{C_1, ..., C_n\} \equiv lcs\{lcs\{C_1, ..., C_{n-1}\}, C_n\} \quad \blacksquare$$