Saying It with Pictures: a logical landscape of conceptual graphs
Kerdiles, G.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
In this chapter, we will consider possible extensions of the simple conceptual graph model. In a search for additional language features and under the control of complexity theory, one cannot escape the well identified tradeoff between expressivity and computational efficiency. Expressive power change is taken here in a broad sense: more than a pure formal notion of symbolic logic comparison, it includes the cognitive impact of considering new language elements that facilitate the representation of information. Indeed, among the different extensions, we will encounter forms of conceptual graph that faithfully depict some characteristics of the represented information such as negative propositions or structured knowledge, while being logically equivalent to the basic language of simple graphs. Continuing on the previous chapter, our guidelines will remain the applications of graphical methods as calculi and the recognition of fragments for which our central benchmark problem, consequence, is tractable.

The language of simple conceptual graphs enables to express positive relational facts between objects with a bit of indeterminacy conveyed by the use of a place holder, the marker '∗'. The first natural extension we consider, is to allow the expressibility of negative facts or in more classical terms, we introduce an atomic negation operator in the language. Chapter 4.1 examines the completeness problem encountered by projection in presence of negated atoms. A syntactical constraint is proposed to define a fragment of simple graphs with atomic negation for which projection is complete and even tractable on guarded forms.

In a second part, Chapter 4.2, we make a large step to a language of graphs as expressive as classical first-order logic. Our perspective is already influenced by the known undecidability of our benchmark problems in FOL, thus we follow a path that moves us away from efficiency considerations and instead, we study the possibility of combining two graphical methods in one complete calculus: projections and classical tableau decomposition rules. Such a generic and modular proof method for the whole language opens the possibility of examining particular tunings of the components in order to suit the needs of intermediate fragments.
Chapter 4. Richer pictures

Our final language takes us back to the cornerstone of this thesis, existential conjunctive FOL, but with a superstructure of hierarchical networks. In this setup, the information conveyed by a graph is partitioned into smaller local pieces of information. The representation of localised - or contextualised - information has been recognised as a need in artificial intelligence. Nested graphs, graphs whose nodes are themselves graphs, follow this principle of locality.

4.1 Atomic negation

Describing the world, one may not only want to provide a positive image! The possibility of expressing negative facts enriches the range of our language. Even in the case of a finite situation in which negative information can be left implicit as it can be extracted from a complete picture of the positive facts, the virtue of a negation operator can be defended in terms of conciseness.

4.1.1 Polarised simple graphs

Syntactically, the most obvious way of discriminating positive facts from negative ones is to attribute a colour, a sign, to every relation occurrence:

4.1.1. Definition. [Polarised simple graph] A polarised simple conceptual graph (PSCG) \( G = (R, C, E, label, co, sign) \) over a signature \( E \) is a simple conceptual graph such that every relation vertex is signed: i.e., \( G = (R, C, E, label, co) \) is a simple conceptual graph and \( sign \), a function from \( R \) to \( \{+,-\} \).

To avoid some overloading of graphics, positive relation nodes are represented without explicit sign whereas negative ones have a label preceded by a "-".

![A polarised simple conceptual graph](image)

Figure 4.1: A polarised simple conceptual graph

4.1.2. Example. The graph drawn in Figure 4.1 represents the information 
\[ c3(a) \land c1(a) \land \exists x(c1(x) \land c2(x) \land R(x, a) \land \exists y(c1(y) \land R(y, x) \land \neg P(y, x))) \]
4.1. Atomic negation

There is one exception to the finiteness of simple graphs: the absurd graph. By opposition to the empty graph, which is satisfied by any structure, the absurd graph represents the conjunction of all possible (finite) polarised simple graphs over a given language signature.

4.1.3. Definition. [Absurd graph] $G_\perp$, the absurd graph, is defined as the juxtaposition of all PSCGs over $\Sigma$.

### 4.1.1.1 Embedding semantics

Polarised simple graphs are interpreted in the same structures respecting the language signature as the (positive) simple conceptual graphs (Chapter 3.1.3). For a graph to be true in a structure under an assignment, the tuple of objects corresponding to the arguments of a negated relation node must not belong to the interpretation of the considered relation.

4.1.4. Definition. [Truth of a polarised graph] Let $\Sigma = (I, (C, \leq_C), (R, \leq_R), \text{arity})$ be a signature, $G = (C, R, E, \text{label}, \text{co}, \text{sign})$ be a PSCG over $\Sigma$ and $M = (D, F)$ be a $\Sigma$-structure,

- $M, f \models G$ if and only if
  1. $\forall c \in C, f(c) \in F(\text{type}(c))$ and
  2. $\forall r \in R$ such that $\text{label}(r) \in R_\pi$,
     - if $\text{sign}(r) = +$, then $(f(r(1)), \ldots, f(r(n))) \in F(\text{label}(r))$,
     - else ($\text{sign}(r) = -$) it holds that $(f(r(1)), \ldots, f(r(n))) \notin F(\text{label}(r))$.

- $M \models G$ iff there exists an assignment $f$ such that $M, f \models G$.
- $G \models H$ iff $H$ is true in every $\Sigma$-structure in which $G$ is true.

This interpretation of polarised graphs by direct mapping from the representation to the represented is in agreement with the recognised cognitive characteristics of pictures: their faithfulness which enables the formal meaning to match intuition. To be verified in a structure, negated atoms must fit into the holes like pieces of a jigsaw puzzle.

### 4.1.1.2 Positivising

Another way of seeing negative relation nodes is to consider that their label belongs to the vocabulary. To apply the "Positivising" technique (see e.g., [BP83] or [Ben88]) to the polarised fragment, we define a meaning-preserving transformation from negative atoms to positive representations.

4.1.5. Definition. Let $\Sigma = (I, (C, \leq_C), (R, \leq_R), \text{arity})$ be a signature,
• We extend $\Sigma$ to a signature $\Sigma^+ = (I, (C, \leq_C), (\mathcal{R}^+, \leq_{\mathcal{R}^+}), \text{arity}^+)$ such that

1. $\forall P \in \mathcal{R}, P^- \in \mathcal{R}^+$ and $\text{arity}^+(P^-) = \text{arity}^+(P) = \text{arity}(P)$,
2. and $\forall P, Q \in \mathcal{R}, (P \leq_{\mathcal{R}^+} Q$ iff $P \leq_{\mathcal{R}} Q)$ and $(Q^- \leq_{\mathcal{R}^+} P^-$ iff $P \leq_{\mathcal{R}} Q$)

• For a polarised SCG $G = (C, R, E, \text{label}, \text{co}, \text{sign})$ over $\Sigma$, its positive form $G^+ = (C, R, E, \text{label}^+, \text{co})$ is the simple conceptual graph over $\Sigma^+$ obtained from $G$ by replacing the label of every negative relation node by the corresponding newly introduced label:

$\forall r \in R$,

1. if $\text{sign}(r) = +$ then $\text{label}^+(r) = \text{label}(r)$ and
2. (ii) if $\text{sign}(r) = -$ and $\text{label}(r) = P$ then $\text{label}^+(r) = P^-.$

Symmetrically, for any simple conceptual graph $H$ over $\Sigma^+$, $H^-$ is the polarised graph obtained by replacing every relation node labelled with $P^-$ by a relation node labelled with $P$ and negatively signed.

• For a given $\Sigma$-structure $M = (D, [\cdot]_M)$, we define a structure $M^+ = (D, [\cdot]_{M^+})$ over $\Sigma^+$ such that

1. $[\cdot]_{M^+}$ is equal to $[\cdot]_M$ on $I, C$ and $R$ and
2. $\forall P \in \mathcal{R}$ of arity $n$, $[P^-]_{M^+} = D^n \setminus [P]_M$.

Symmetrically, a $\Sigma^+$-structure $N$ is transformed into a $\Sigma$-structure $N^-$ such that

$\forall P \in \mathcal{R}$, $[P]_{N^-} = [P]_N$.

We note that relation symbols of the original signature and newly introduced ones are $\leq_{\mathcal{R}^+}$-incomparable.

On one hand, back and forth translations of graphs preserve all information: the sign of a relation node is just internalised in the label and vice-versa. On the other hand, structure transformations call for a more careful analysis which presents the premisses for the soundness but incompleteness of projection in the polarised fragment (further developed in Chapter 4.1.2).

4.1.6. FACT. Let $M$ be a $\Sigma$-structure, $N$ be a $\Sigma^+$-structure, $G$ be a polarised simple graph over $\Sigma$ and $H$ be a simple graph over $\Sigma^+$,

1. $G^{+-} = G$ and $H^{++} = H$
2. $M^+$ is a proper $\Sigma^+$-structure and $N^-$ is a proper $\Sigma$-structure
3. $M \models G \Rightarrow M^+ \models G^+$ and $N \models H \Rightarrow N^- \models H^-$

provided that $\forall P \in \mathcal{R}$, $[P]_N \cap [P^-]_N = \emptyset$ and $H^-$ is not absurd.
4.1. Atomic negation

4. \( M^{+-} = M \), but there exists a \( \Sigma^+ \)-structure \( N \) such that \( N^{-+} \neq N \).

Proof:

1. Graph transformations purely concerns the layout by gathering (or splitting) two kinds of labels attached to relation nodes: sign and relation symbol.

2. To prove that \( M^+ \) is a \( \Sigma^+ \) structure, it is sufficient to verify that \( \forall P, Q \in \mathcal{R}, \) if \( P \leq_R Q \) then \( [Q^-]_{M^+} \subseteq [P^-]_{M^+} \). Indeed, the interpretation of the remaining symbols is preserved from \( M \). Let \( P, Q \in \mathcal{R} \), of arity \( n \) such that \( P \leq_R Q \). \( M \) is a \( \Sigma \)-structure, thus, \( [P]_M \subseteq [Q]_M \). As \( [Q^-]_{M^+} = D^n \setminus [Q]_M \) and \( [P^-]_{M^+} = D^n \setminus [P]_M \), it directly follows that \( [Q^-]_{M^+} \subseteq [P^-]_{M^+} \).

Symmetrically, the ordering on relation symbols is directly preserved by the transformation from \( N \) to \( N^- \) as \( [P]_{N^-} = [P]_N \).

3. \( M^+ = (D, [\cdot]_{M^+}) \) is a \( \Sigma^+ \)-structure of a special form: given a relation symbol \( P \in \mathcal{R} \) of arity \( n \) and \( d \in D^n \), either \( d \in [P]_{M^+} \) or \( d \in [P^-]_{M^+} \);

The classical two-valuation is forced.

Assume for an assignment \( f \) that \( M, f \models G \). In \( G^+ \), every atomic compound with a relation node labelled with a symbol in \( \mathcal{R} \) must be satisfied by \( M \) under \( f \); hence, by \( M^+ \) under \( f \) as the interpretation of relation symbols in \( \mathcal{R} \) is preserved by the structure transformation.

On the other hand, any atomic compound \( r(\tilde{t}) \) in \( G^+ \) with \( \text{label}(r) = P^- \) corresponds to an atomic compound \( r'(\tilde{t}) \) in \( G \) with \( \text{label}(r') = P \) and \( \text{sign}(r') = - \). \( M, f \models G \), thus \( f(\tilde{t}) \notin [P]_M \). Hence, \( f(\tilde{t}) \in [P^-]_{M^+} \).

Symmetrically, the interpretation of positive relation symbols is also preserved (i.e., \( \forall P \in \mathcal{R}, [P]_{N^-} = [P]_N \)) and as overvaluation is avoided (i.e., it is not the case that \( \tilde{t} \in ([P]_N \cap [P^-]_N) \) for any \( \tilde{t} \) and \( P \)), it follows that \( \tilde{t} \in [P^-]_N \) implies that \( \tilde{t} \notin [P]_{N^-} \).

4. The chosen structure transformations are based on the preservation of the positive information:

(i) \( \tilde{t} \in [P]_M \Rightarrow \tilde{t} \in [P]_{M^+} \Rightarrow \tilde{t} \in [P]_{M^{+-}} \) and

(ii) \( \tilde{t} \notin [P]_M \Rightarrow \tilde{t} \notin [P]_{M^+} \Rightarrow \tilde{t} \notin [P]_{M^{+-}} \)

Hence, \( M^{+-} = M \).

On the other hand, “undervaluation” in a \( \Sigma^+ \)-structure is transformed into negative information and “overvaluation” into positive one; hence modifying the original structure:

(i) if both \( \tilde{t} \notin [P]_N \) and \( \tilde{t} \notin [P^-]_N \), then \( \tilde{t} \notin [P]_{N^-} \), thus \( \tilde{t} \in [P^-]_{N^{+-}} \)

(ii) if both \( \tilde{t} \in [P]_N \) and \( \tilde{t} \in [P^-]_N \), then \( \tilde{t} \in [P]_{N^-} \), thus \( \tilde{t} \notin [P^-]_{N^{+-}} \)

Hence, there exists a \( \Sigma^+ \)-structure \( N \) such that \( N^{+-} \neq N \).
From the previous facts, follows a first application of the positive transformation of polarised graphs:

4.1.7. THEOREM. For \( G \) and \( H \) two polarised simple conceptual graphs over \( \Sigma \),

\[
G^+ \subseteq H^+ \implies G \subseteq H 
\]

**Proof:** assume that any \( \Sigma^+ \) structure that satisfies \( G^+ \) also satisfies \( H^+ \). Let \( M \) be a \( \Sigma \) structure which satisfies \( G \), by the previous fact (Fact 4.1.6(3)), \( N^+ \) is a \( \Sigma^+ \) structures which satisfies \( G^+ \). Thus, by assumption, \( N^+ \models H^+ \). Thus, again by Fact 4.1.6(3), \( N^+ \models H^+ \). Hence, by Fact 4.1.6(1 & 4), \( N \models H \)

Unfortunately the reciprocal of the previous theorem is not true: we will see in Chapter 4.1.2 that the undervaluation case, in the proof of Fact 4.1.6(4), forbids the use of the minimal model of a polarised graph transformed into positive form as a fair representant of the information conveyed by the original polarised graph.

Our next step in manipulating the polarised fragment has become clear. Polarised graphs resemble their simple conceptual graph ancestors: drawings have the same bipartite network structure and are interpreted by the same kind of direct mapping to models. The sole difference resides in the labelling, it is thus legitimate to consider the possibility of solving the consequence decision problem by a special projection taking care of the new kind of labels.

4.1.1.3 Projection

The projection calculus corresponds to the evaluation of a source graph in the canonical model of a target graph. We have seen in the simple graph fragment that the method is complete if the target graph is isomorphic to its canonical model, or in conceptual graph terms, normalised. It seems natural to extend the projection algorithm to polarised simple graphs, by just adding for negated nodes, a constraint on labels which is symmetrical to the usual one for positive relations: while the neighbourhood of a relation node must be preserved in the mapping, a positive relation node is mapped onto another positive one which has a more specialised label, whereas a negative relation node is mapped onto another negative one with a more general label.

4.1.8. DEFINITION. [Projection for PSCGs] Let \( G = (C, R, E, \text{label}, \text{co}, \text{sign}) \) and \( H = (C', R', E', \text{label}', \text{co}', \text{sign}') \) be two polarised simple conceptual graphs over a signature \( \Sigma = (I, (C, \leq_C), (R, \leq_R), \text{arity}) \). A projection from the source \( H \) to the target \( G \) is a mapping \( \pi : C' \cup R' \to C \cup R \) such that:
4.1. Atomic negation

- $\forall c \in C', \pi(c) \in C$ and $\text{type}(\pi(c)) \leq c \text{type'}(c)$

- $\forall c \in C'$, if $\text{marker'}(c) \in I$ then $\text{marker}(\pi(c)) = \text{marker'}(c)$

- $\forall r \in R', \pi(r) \in R$ and
  - if $\text{sign}'(r) = +$, then $\text{sign}(\pi(r)) = +$ and $\text{label}(\pi(r)) \leq R \text{label'}(r)$,
  - else $\text{sign}(\pi(r)) = -$ and $\text{label'}(r) \leq R \text{label}(\pi(r))$.

- $\forall (r, c, i) \in E', (\pi(r), \pi(c), i) \in E$

- $\forall (c_1, c_2) \in c0', \pi(c_1) = \pi(c_2)$

Figure 4.2: A projection with polarised graphs

4.1.9. EXAMPLE. In Figure 4.2, the source graph representing "$c3(a) \land c1(a) \land \exists x(c1(x) \land c2(x) \land R(x, a) \land \exists y(c1(y) \land R(y, x) \land \neg P(y, x))$" is a logical consequence of the target graph representing "$c3(a) \land \neg P'(a, a) \land R(a, a)$", given the information that $c3$ is a subconcept of $c1$ and $c2$ and that the relation $P$ is a subrelation of $P'$.

Equivalently presented, there is a projection between two polarised simple graphs iff there is a projection between their respective positive representation:

4.1.10. FACT. For two polarised simple conceptual graphs $G$ and $H$ over $\Sigma$, there is a projection from $H$ to $G$ if and only if there is a projection, with respect to the orders in $\Sigma^+$, from $H^+$ to $G^+$. 
Chapter 4. Richer pictures

Proof: it is immediate to verify that the projection-constraint on negative relation nodes is transferred to the signature order over the negated relational vocabulary.

4.1.11. COROLLARY (SOUNDNESS OF PROJECTION FOR POLARISED GRAPHS). For $G$ and $H$ two polarised simple conceptual graphs over $\Sigma$,

if there exists a projection from $H$ to $G$, then $G \subseteq H$

Proof: if there exists a projection from $H$ to $G$, then there exists a projection from $H^+$ to $G^+$ by Fact 4.1.10. Thus $G^+ \subseteq H^+$ by the soundness of projection on simple conceptual graphs (Theorem 3.2.6). Hence, $G \subseteq H$ by Theorem 4.1.7.

Soundness is not sufficient. We also aim at a complete method, but atomic negation has introduced a form of disjunctive information which prevents the “all in one mapping” method to capture all consequences.

4.1.2 Insufficiencies of projection

We already know from the fragment of positive simple graphs, that the target graph is required to appear in normal form (i.e., that distinct concept nodes representing a single object must be merged). Fortunately, this normal form can be obtained in polynomial time.

Normalisation put aside, there is still a source of incompleteness which is linked to the fact that the calculus has to cope with tautologies of the form “$P(\ddot{x}) \lor \neg P(\dddot{x})$”.

Take, for instance, the sentence $\varphi_1 = P(a) \land R(a, b) \land R(b, c) \land \neg P(c)$. It entails the sentence $\varphi_2 = \exists x \exists y (P(x) \land \neg P(y) \land R(x, y))$ because, from $\varphi_1$, either it holds that $P(b)$ and in this case the pair of variables $(x, y)$ can be unified with $(b, c)$, or it holds that $\neg P(b)$ and in this case $(x, y)$ can be unified with $(a, b)$. From the viewpoint of model-checking, every model of $\varphi_1$ satisfies either $P(b)$ or $\neg P(b)$. However, the conceptual graph formalism does not allow for an explicit representation of this disjunctive information and there is no projection from

The following second counter-example to the completeness of projection involves a target graph where positive and negative information are dispatched

![Diagram of Conceptual Graphs]
in different connected compounds: $R(a, b) \land \neg R(c, d) \models_{FOL} \exists x \exists y \exists z (R(x, y) \land \neg R(y, z))$ whereas the graph cannot be projected onto the graph.

To take an alternative perspective on the problem, we can observe projection through the previously presented positive transformation of polarised graph (Chapter 4.1.1.2). We could be tempted to apply the reciprocal reasoning to the proof of Theorem 4.1.7:

consider $G^+$ and $H^+$ the positive forms of the respective polarised graphs $G$ and $H$. Assume that $G \subseteq H$ and let $N$ be the minimal model of $G^+$.

By assumption, $N^- \models H$, thus, again by Fact 4.1.6(3), $N^+ \models H^+$.

Unfortunately, $N^+$ is not a sub-model of the original structure $N$, hence we cannot conclude that $N \models H$.

Stronger, we can conclude from the previous counter examples that it does not hold that for $G$ and $H$ two polarised simple conceptual graphs, $G \subseteq H \Rightarrow G^+ \subseteq H^+$.

So, are we forced to completely abandon our project of mapping polarised graphs? No, as we will see, constraints on the interaction of negations and existential quantifiers allow us to correctly base our reasoning technique on projection and even in a tractable way under guarded conditions.

4.1.3 Discriminated polarised simple graphs

We have already noticed the relevance of the notion of connected compound to the projection calculus for positive simple conceptual graphs: a connected compound is a piece of information which is independent from the remaining parts. Therefore, if each connected compound of a source graph can be projected on a target graph, then we obtain a projection of the whole source graph by taking the union of all these “small” projections. This idea can be applied to polarised graphs in order to treat separately positive and negative pieces of information.

A polarised simple graph is called discriminated if none of its relation nodes shares a concept node neighbour with a relation node of the opposite sign.

4.1.12. DEFINITION. [Discriminated simple graphs] A polarised simple graph $G = (R, C, E, label, co, sign)$ is discriminated if $\forall r \in R$ such that $sign(r) = +$ it holds that $\forall (r, c, i) \in E, (r', c, j) \in E$ implies that $sign(r') = +$.

Because splitting an instantiated concept node into many copies preserves the meaning of a graph (the distinct copies denote the same object in a model),
Chapter 4. Richer pictures

Figure 4.3: A discriminated graph

A polarised graph can be transformed into an equivalent discriminated graph if all paths of edges between two relation nodes of opposite sign include some instantiated concept nodes. For instance, the graph in Figure 4.3 is equivalent to the following non-discriminated graph:

![Non-discriminated graph](image)

Obviously, in terms of formulae, a discriminated graph corresponds to a conjunction of two sentences, one with only positive occurrences of relations and the other one with only negated occurrences of relations.

4.1.4 Completeness and tractability

4.1.4.1 Discriminated graphs and the splitting of labour

Before getting into the details of the desired completeness proof for the discriminated fragment, let us come back to the simple proof scheme proposed in Chapter 4.1.2. Failing to show that $G \subseteq H \Rightarrow G^+ \subseteq H^+$, we identified a culprit as a missing identity $N^+ = N$, or more precisely, a missing inclusion $N^+ \subseteq N$, as the satisfaction of a positive graph is preserved under model expansion.

In this section, we can still not prove that the minimal model $M_{G^+}$ of $G^+$ is preserved under $-+$ transformations, but we can use the fact that the discrimination in the conclusion graph, $H^+ = H^+_{pos} \oplus H^+_{neg}$, allows us to consider independently positive and negative information and gather the results at the end. We will prove that if the conclusion $H$ contains either only positive facts or only negative facts, then we can exhibit a sub-model $N \subseteq M_{G^+}$ constructed from $(M_{G^+})^-$ such that $N \models H^+$. Hence $(G \subseteq H) \Rightarrow (M_{G^+} \models H^+_{pos} \oplus H^+_{neg}) \Rightarrow (G^+ \subseteq H^+)$. We have seen that the transformation from a $\Sigma^+$-structure $N$ to the $\Sigma$-structure $N^-$ favours truly positive knowledge: $\forall P \in R$, $[P]_{N^-} = [P]_N$. To restore the balance, we shall use a second transformation which favours truly negative knowledge:
4.1. Atomic negation

4.1.13. DEFINITION. \( \sqrt{\text{-transformation}} \) Given a signature \( \Sigma \) and a \( \Sigma^+ \)-structure \( N = (D, [\cdot]_N) \), we define \( N^\sqrt{\cdot} = (D, [\cdot]_{N^\sqrt{\cdot}}) \) such that \( \forall P \in \mathcal{R} \text{ of arity } n, \forall \vec{t} \in D^n, \)

\[
\vec{t} \not\in [P]_{N^\sqrt{\cdot}} \text{ iff } \vec{t} \in [P^-]_N
\]

4.1.14. FACT. Given a signature \( \Sigma \) and a \( \Sigma^+ \)-structure \( N \),

1. \( N^\sqrt{\cdot} \) is a proper \( \Sigma \)-structure
2. \( \forall P \in \mathcal{R}, \text{ if } [P]_N \cap [P^-]_N = \emptyset, \text{ then } [P]_N \subseteq [P]_{N^\sqrt{\cdot}} \)
3. \( N \models H \Rightarrow N^\sqrt{\cdot} \models H^- \) provided that \( \forall P \in \mathcal{R}, [P]_N \cap [P^-]_N = \emptyset \) and \( H^- \) is not absurd.

Proof:

1. We must verify that the chosen ordering on relation symbols is preserved by the \( \sqrt{\cdot} \)-transformation. For \( P \leq_R Q \), suppose that there exists \( \vec{t} \in [P]_{N^\sqrt{\cdot}} \) such that \( \vec{t} \not\in [Q]_{N^\sqrt{\cdot}} \). By Definition 4.1.13, \( \vec{t} \in [Q^-]_N \) and, as \( N \) is a proper \( \Sigma^+ \)-structure, \( [Q^-]_N \subseteq [P^-]_N \), thus \( \vec{t} \in [P^-]_N \). Again, by Definition 4.1.13, it follows the contradiction: \( \vec{t} \not\in [P]_{N^\sqrt{\cdot}} \).

2. If \( N \) does not contain any overvaluation, then truly positive facts are preserved by the \( \sqrt{\cdot} \)-transformation: suppose that \( \vec{t} \in [P]_N \) and \( \vec{t} \not\in [P]_{N^\sqrt{\cdot}} \), then, by Definition 4.1.13, \( \vec{t} \in [P^-]_N \). Thus, from \( [P]_N \cap [P^-]_N = \emptyset \), it follows the contradiction that \( \vec{t} \not\in [P]_N \).

3. We have just proved that positive information is preserved by the transformation. On the other hand, \( \forall P \in \mathcal{R} \text{ of arity } n, (D^n \setminus [P]_{N^\sqrt{\cdot}}) \cap [P^-]_{N^\sqrt{\cdot}} = \emptyset \) for any relation symbol \( P \), thus the satisfaction of negative facts is also preserved by the transformation.

To prove the completeness theorem, i.e. \( (G \subseteq H) \Rightarrow (G^+ \subseteq H^+) \), we have learned from Chapter 2 and Chapter 3 that it sufficient to prove that \( G \subseteq H \) implies that \( M_{G^+} \models H^+ \), as \( G^+ \) and \( H^+ \) are simple conceptual graphs (over an extended vocabulary). Let us briefly recall the construction of the minimal model of the simple graph \( G^+ \).

Without loss of generality, we assume that \( G \) is not absurd, thus \( [P]_{M_{G^+}} \cap [P^-]_{M_{G^+}} = \emptyset \) for any relation symbol \( P \). The universe of \( M_{G^+} \) consists in the set of concept nodes in the normal form of \( G^+ \), or equivalently, the set of concept node labels in the graph obtained from \( G^+ \) by replacing each coreference class...
by a witness. For any relation symbol \( P \), \( [P]_{M_G^+} = \{ i/\exists Q \leq R P, Q^{-i} \in G^+ \} \) and \( [P^{-}]_{M_G^+} = \{ i/\exists Q \geq R P, Q^{-i} \in G^+ \} \).

It directly follows from the model construction that the minimal model of \( G^+ \) satisfies \( G \) under both transformations \(-\) and \( \check{} \).

4.1.15. FACT.

\[ (M_{G^+})^\check{} \models G \text{ and } (M_{G^+})^- \models G \]

Proof: \( M_{G^+} \models G^+ \) by construction (Fact 3.2.4). As \( G \) is not absurd, \( M_{G^+} \) is not over-valuated. Thus, \( (M_{G^+})^- \models G^+^- \) (by Fact 4.1.6), \( (M_{G^+})^\check{} \models G^+^- \) (by Fact 4.1.14) and \( G^+^- = G \) (by Fact 4.1.6).

We can now employ the \( \check{} \) and \(-\) transformations to prove the following lemma:

4.1.16. LEMA (COMPLETENESS WITH A UNIFORMLY SIGNED CONCLUSION).

Let \( \Sigma \) be a signature, \( G \) be a non absurd polarised simple conceptual graph over \( \Sigma \) and \( H \) be a polarised simple conceptual graph whose relation nodes all have the same sign, it holds that

\[ (G \sqsubseteq H) \Rightarrow (G^+ \sqsubseteq H^+) \]

Proof: Assume that \( G \sqsubseteq H \).

- Let \( H \) be negative (i.e., every relation node in \( H \) is negatively signed or, equivalently, every relation node in \( H^+ \) is labelled with a negatively signed relation symbol.

We have just proved that \( (M_{G^+})^\check{} \models G \).

Hence, by assumption \( (M_{G^+})^\check{} \models H \).

Thus, by Fact 4.1.6(3), \( (M_{G^+})^\check{} + \models H^+ \)

Furthermore, we can verify that \( \forall P \in R, [P^-]_{M_G^+} = [P^-]_{(M_{G^+})^\check{}} + \) :

\( \check{t} \in [P^-]_{M_G^+} \iff \check{t} \notin [P]_{(M_{G^+})^\check{}} \) (by Definition 4.1.13)

\( \check{t} \in [P]_{(M_{G^+})^\check{}} + \) (by Definition 4.1.5).

So, \( (M_{G^+})^\check{} + \) and \( M_{G^+} \) coincide on the interpretation of negated relation symbols.

Furthermore, as \( H \) be negative, we can safely eliminate from the model the sets of tuples in the interpretation of positive relation symbols:

let \( N \) have the same universe as \( (M_{G^+})^\check{} + \) and \( \forall P \in R, [P]_N = \emptyset \) and \( [P^-]_N = [P^-]_{(M_{G^+})^\check{}} + \). It holds that \( N \models H^+ \) and that \( N \subseteq M_{G^+} \).

Hence \( M_{G^+} \models H^+ \) as the satisfaction of a simple graph is preserved under model expansion.
4.1. Atomic negation

Let $H$ be positive (it is a simple conceptual graph). Similarly to the negative case, but using the $-$ transformation, we can construct a submodel of the minimal model of $G^{+}$ that satisfies $H^{+}$:

$$(M_{G^{+}})^{-} \models G \Rightarrow (M_{G^{+}})^{-} \models H \Rightarrow (M_{G^{+}})^{-} \models H^{+}$$

Let $N$ have the same universe as $(M_{G^{+}})^{-}$ and $\forall P \in \mathcal{R}, [P]_{N} = [P^{-}]_{(M_{G^{+}})^{-}}$ and $[P^{-}]_{N} = \emptyset$. It holds that $N \models H^{+}$ and that $N \subseteq M_{G^{+}}$. Hence, $M_{G^{+}} \models H^{+}$.

4.1.4.2 Projection completeness

We are now ready to prove the completeness of projection from a discriminated graph to a polarised one in normal form. We can apply the lemma to each part of a discriminated graph:

4.1.17. COROLLARY. For $G$ and $H$ two polarised simple conceptual graphs over $\Sigma$ such that $G \neq G_{\perp}$ and $H$ is discriminated,

$$G \subseteq H \Rightarrow G^{+} \subseteq H^{+}$$

Proof: $H$ is discriminated, so we can equivalently rewrite it as the juxtaposition of two polarised simple graphs $H_{pos}$ and $H_{neg}$ such that all relation nodes in $H_{pos}$ are positive and all relation nodes in $H_{neg}$ are negative; $H = H_{pos} \oplus H_{neg}$.

Assume that $G \subseteq H$. From Lemma 4.1.16, we have $\exists f, g$ assignments such that $M_{G^{+}}, f \models H_{pos}^{+}$ and $M_{G^{+}}, g \models H_{neg}^{+}$. Hence, $M_{G^{+}}, (f \cup g) \models H_{pos}^{+} \oplus H_{neg}^{+}$ as $f$ and $g$ coincide on the interpretation of individual marker (by definition of an assignment) and $H_{pos}^{+}$ and $H_{neg}^{+}$ do not share any existential concept node. Hence $M_{G^{+}} \models H^{+}$.

The purpose of normalisation is twofold: first, as for positive graphs, concept nodes representing a single object are merged in order to obtain a graph isomorphic to its minimal model (if such a model exists). Furthermore, if the graph contains contradictory information, then it is replaced by the absurd graph. This operation guarantees that any graph can be projected onto the normal form of a graph representing contradictory information (more specifically, any graph can be mapped onto its copy which is part of the absurd graph).

4.1.18. DEFINITION. [Normal form of a polarised graph] $\text{Norm}(G)$, the normal form of a PSCG $G = (C, R, E, label, co, sign)$ is defined in two steps as follows:
1. Let $G' = (C', R, E', \text{label}', \text{sign}')$ be the PSCG obtained from $G$ by merging all occurrences of concept nodes having the same individual marker and, for each coreference equivalence class, merging all elements of the class. After merging two nodes, the resulting one is labelled with the meet of the concept label of the original nodes.

2. If in $G'$ there are two relation nodes $r$ and $r'$ in $R$ such that $\text{label}'(r') = p$, $\text{sign}'(r') = +$, $\text{label}'(r) = q$, $\text{sign}'(r) = -$, $p \leq_R q$ and $\forall (r', c, i) \in E, (r, c, i) \in E$, then $\text{Norm}(G) = \bot$; otherwise $\text{Norm}(G) = G'$.

4.1.19. Theorem (Completeness of Projection). For a polarised graph in normal form $G$ and a discriminated graph $H$, $G \sqsubseteq H$ iff there exists a projection from $H$ to $G$.

Proof: we have already considered the “if direction” in Corollary 4.1.11. Reciprocally, if $G$ is absurd, then $H$ is a subgraph of $G$ (as $G$ is in normal form). Hence, there is projection from $H$ to $G$. Otherwise, $G^+ \sqsubseteq H^+$ by Corollary 4.1.17 and building on the completeness of projection for simple conceptual graphs (Theorem 3.2.6), we can conclude that there is a projection from $H^+$ to $G^+$. Hence there is a projection from $H$ to $G$ by Fact 4.1.10.

4.1.4.3 Tractability harvest

We can directly benefit from the complete transposition of discriminated polarised graphs into the simple conceptual graph fragment.

4.1.20. Corollary. • The empty graph is the sole valid polarised graph.
• Deciding satisfiability of a polarised graph is in polynomial time.
• $G \sqsubseteq H$ is NP-complete, for $G$ and $H$ two polarised graphs over a common signature and such that $H$ is discriminated.

Proof:
• A structure with empty universe can only satisfy the empty graph which is by definition also satisfied by any structure (no required embedding).
• As in the corresponding textual fragment (Proposition 2.6.2), we can verify that no opposite relation nodes share the same arguments by a simple check through a polarised graph: a brutal procedure that goes through the whole graph for each relation node requires a quadratic time.
• By Theorem 4.1.17, the polarised consequence problem with a discriminated conclusion is in NP. Indeed, the translation of a polarised graph into its positive
Atomic negation

Form is only a linear renaming. The NP lower bound is the one of subsumption in simple graphs as any simple graph is a polarised one.

We can exploit even further the correspondence to (positive) simple graphs:

4.1.21. Definition. [Guarded discriminated polarised graphs] A discriminated polarised graph is guarded if it is the result of juxtaposing two polarised graphs \( G \) and \( H \) such that (i) all relation nodes in \( G \) are positive, (ii) all relation nodes in \( H \) are negative and (iii) \( G \) and \( H \) are guarded simple graphs if we make abstraction of the signs.

4.1.22. Corollary. Given two polarised graphs \( G \) and \( H \) over a common signature, \( G \subseteq_{NCG} H \) is decidable in polynomial time if \( H \) is guarded and discriminated.

Proof: Immediate consequence of Theorem 4.1.17 and Theorem 3.3.7 as the transformation of the problem into the (guarded) fragment of simple graphs is polynomial.

4.1.5 Concluding remarks

The lesson we can draw from this first extension of the simple graph fragment is that a very simple additional language feature such as atomic negation can already compromise the enterprise of basing reasoning on a single picture comparison.

Nevertheless, we have been able to identify a constraint on the disposition of negations in a picture which defines a fragment of polarised graphs where the projection technique safely applies. Discriminated polarised graphs are representations built from the juxtaposition of proper polarised graphs which contain either only positive facts or only negative ones.

Furthermore, by transposing back the problem into the guarded fragment of simple graphs, we have reinforced the relevance of tree-like structures for the computational tractability of the consequence problem.

We should note that to our disadvantage, disjunctive forms that, by nature, do not lend themselves easily to pictorial representations, are among the most extensively decorticated fragments of FOL in the literature. For instance, languages based on clausal forms, such as Horn-fragments, have proved their computational efficiency ([BGG97] provides a very extensive survey of complexity results and bibliographic references for such clausal languages). Also in database theory, algorithmic solutions to the query containment problem in different languages of conjunctive queries including forms of negation have been proposed; see e.g. [LS93], [LMSS93], [LS95] or [FTU99].
Back to the graphs, we may not have yet played all our cards on projection with atomic negation: the implicit disjunctive information hidden in polarised representations could possibly be handled by combining alternative simultaneous projections.

This idea of decomposing the proof of a disjunction into parallel subproofs is also the core of a classical type of calculus in logic, semantic tableaux. Leaving open the question of the applicability of simultaneous projections, we will examine how tableaux can be combined to projections in a wider graph fragment.

4.2 Full classical negation

This section is taken up with a rather different angle than the previous ones. Indeed, by considering a conceptual graph language which is equivalent to FOL, our goal cannot be tractability (and even decidability) anymore, but consistently with the former fragments, our focus can still be directed towards graphical methods that naturally apply to the pictorial representations.

We will propose a proof method interlacing decomposition steps by tableau rules\(^1\) applied to complex graphs and projection steps on sufficiently simple subgraphs. For its generic character, the calculus is not aimed as such at a direct efficient implementation, but rather at being easily reinforced by heuristics that abound in both the tableau and the graph homomorphism literature. Such a fine tuning should be application driven and take place when specific forms of graphs are considered.

It should be emphasised that despite their origins in textual symbolic logics, tableaux offer an overwhelming position to a graphical aspect: trees serve the purpose of collecting in a compact way the deterministic construction of alternative (counter) models.

4.2.1 Conceptual Graphs

In a preliminary stage, we define a conceptual graph representation of negation inspired by Peirce's notation in existential graphs (e.g., [Rob73]) and also very similar to the box style of DRS syntax (e.g., [KR93]).

4.2.1.1 Negation box syntax

The negation of a graph is represented by a closed line surrounding the graph (or in Peirce's terms, a line cutting the portion of the sheet where the graph is drawn). Furthermore, closed lines are not crossing each other. Hence they define

\(^1\)See e.g., [Snu68] for a very pleasant presentation of the method or [Fit90] for an implementation oriented introduction. The proceedings of the annual tableau conference also form a rich knowledge base of application oriented techniques.
4.2. Full classical negation

an order, a tree structure, on the nesting of delimited portions of the assertion sheet. Coreference links are required to respect this order.

4.2.1. **DEFINITION.** [Conceptual graphs] A Conceptual Graph (CG) over a signature $\Sigma$, is a set of entries strengthened by a coreference equivalence relation on existential concept nodes, $G = \{G_1, \ldots, G_{n(1 \leq n)}\}, co$, such that the following holds:

1. an entry is
   (a) either simple if it is a simple conceptual graph (possibly the empty graph $G_0$) over $\Sigma$:
      
      $$G' = (R', C', E', label', co') \text{ where } co' \text{ is the identity relation on } C'_*;$$
   (b) or boxed if it is of the form $G'$ where $G'$ is a CG over $\Sigma$

2. $G_1$ is simple.

3. if two concept nodes, $c$ and $c'$, are labelled with the same object name, $m \in I$, or are co-equivalent, then they are also labelled with the same concept type $t \in C$

4. the coreference relation $co$ extends the coreference relation of the $n$ entries: $c \equiv_{co} c'$ implies that
   (a) either the two nodes are already coreferent in a single entry $G_{i(1 \leq i \leq n)}$ in which they occur
   (b) or there exists an existential concept node $c''$ in the simple entry $G_1$ such that $c \equiv_{co} c' \equiv_{co} c''$.

$CG(\Sigma)$ denotes the set of all finite conceptual graphs with respect to the signature $\Sigma$. Abusing the former notation for simple graphs, we call $G_0$ the empty conceptual graph: viz., the graph whose single entry is the empty SCG.

4.2.2. **EXAMPLE.** To emphasise the nesting of boxes, we use two different background tints for the subgraphs surrounded by an odd number of boxes and those surrounded by an even number of boxes.

The simple entry is the empty graph. The conceptual graph represents the negation of the information conveyed by the enclosed simple graph:

$$\neg(\exists x (Tx \land Ta \land Ra_x))$$
As in simple graphs, co-reference is represented by a covering set of dashed edges:

This conceptual graph represents

\[ Ta \land Qa \land \neg(\exists x(Tx \land Px)) \land \neg(\exists y(Ty \land Qy \land \neg(Ty \land Ry))) \]

or equivalently (in classical FOL)

\[ Ta \land Qa \land \forall x(Tx \rightarrow \neg Px) \land \forall y((Ty \land Qy) \rightarrow Ry) \]

Note that the three existential concept nodes cannot form a coreference equivalence class as they belong to different entries and cannot find an anchor concept node in the simple entry of the whole CG.

This last graph exemplifies the fact that a finite support for the vocabulary can now directly be expressed in the graph language. Nevertheless, we choose to define CGs over hierarchical signatures to emphasise the role of the simple conceptual graph building block, both syntactically and as part of a CG-calculus.

Symptomatic of the textual representation of conceptual graph pictures, we need additional and somehow overloaded formal definitions to support our set-oriented exposition of the framework.

4.2.3. DEFINITION. [Degree, independent and dominant declarations] For any Conceptual Graph, \( G = (\{G_1, \ldots, G_{n(1\leq n)}\}, co) \).

- \( simple(G) \) denotes the SCG \( G_1 \) and \( complex(G) \) the set of complex entries of \( G \). \( G \) is called simple if it contains no complex entry: \( G = \{simple(G)\} \).

- To facilitate the notations, nodes, edges and labels are recursively uplifted from entries to the whole graph:

\[ \forall X \in \{C, R, E, label\}, X_G = \bigcup_{1 \leq i \leq n} X_{G_i} \]

where for \( X \in \{C, R, E, label, co\} \), on a complex entry \( G'' = [G'] \), \( X_{G''} = X_{G'} \) and on a SCG \( G' = (C', R', E', label', co') \), \( X_{G'} = X' \).
4.2. Full classical negation

- Degree of a conceptual graph: if $G$ is simple then $\deg(G) = 0$, else $\deg(G)$ is the sum of the degrees of the complex entries in $G$. The degree of a complex entry $G'$ is $1 + \deg(G')$.

- Declarations are existential concept nodes in the simple entry of a conceptual graph. They are partitioned into:

  1. independent declarations: $IDec(G)$ is the set of existential concept nodes in the simple entry of $G$, which are not coreferent to other distinct concept nodes.

     $$IDec(G) = \{ c \in simple(G) / \text{marker}^G(c) = * \& \text{class}_{co}(c) = \{c\} \}$$

  2. dominant declarations: the declarations which bind other concept nodes deeper in the nesting of entries.

     $$DDec(G) = \{ c \in simple(G) / \text{marker}^G(c) = * \& \text{class}_{co}(c) \neq \{c\} \}$$

     $$Dec(G) = IDec(G) \cup DDec(G) = \{ c \in simple(G) / \text{marker}^G(c) = * \}$$

- It is convenient to transform an entry into a proper conceptual graph. $Graph(G_1) = (\{G_1\}, co_{G_1})$ and $graph(G_{i_1 < i < n}) = (\{G_1, G_i\}, co_{G_i})$.

When it is clear from the context, we often lighten notations by writing $G$ instead of $\text{graph}(G)$.

4.2.1.2 Interpretation of negated graphs

Extending the embedding semantics of simple graphs, it is natural to interpret a negated graph by testing whether its content can be mapped on part of a model. We first need to define a way to convey recursively in the nesting of negation boxes successful partial mappings: assignments. A partial assignment of a conceptual graph in a structure extends the interpretation function of individual markers and relation symbols, but it can remain undefined for some existential concept nodes:

4.2.4. Definition. [Partial Assignments]

For a signature $\Sigma = (\mathcal{I}, (C, \leq_C), (\mathcal{R}, \leq_{\mathcal{R}}), \text{arity})$, a CG $G$ over $\Sigma$ and a $\Sigma$-structure $M = (D, F)$, a partial assignment of $G$ in $M$ is a partial function from the concept nodes in $G$ to $D$ such that $\forall c, c' \in C_G$; both:

1. If $\text{marker}(c) \in \mathcal{I}$ then $f(c) = F(\text{marker}(c))$ and $f(c) \in F(\text{type}(c))$ (hence, $f(c)$ is defined on all instantiated concept nodes)

2. If $c \equiv_{co} c'$ and $f(c)$ is defined, then it holds that $f(c') = f(c) \in F(\text{type}(c))$. 
The empty assignment of \( G \) in \( M \) is the unique partial assignment, \( f \), of \( G \) in \( M \) such that \( \forall c \in C_G, \text{marker}(c) = * \) implies that \( f(c) \) is undefined.

The meaning of a conceptual graph can now be recursively defined from the meaning of simple sub-graphs:

**4.2.5. Definition. [Truth of a CG]** Let \( \Sigma = (\mathcal{I}, (\mathcal{C}, \leq_C), (\mathcal{R}, \leq_R), \text{arity}) \) be a signature, \( G = (\{G_1, ..., G_{n(1 \leq n)}\}, \text{co}) \) be a CG over \( \Sigma \) and \( M = (D, F) \) be a \( \Sigma \)-structure. Let \( f \) be a partial assignment of \( G \) in \( M \).

- \( M, f \models G \) iff there exists a partial assignment \( g \) of \( G \) in \( M \) such that:
  1. \( g \) extends \( f \) and the restriction of \( g \) to \( \text{simple}(G) \) is an assignment of \( \text{simple}(G) \) in \( M \) (cf. Definition 3.1.7) and
  2. \( M, g \models \text{simple}(G) \) (cf. Definition 3.1.8) and
  3. if \( n > 1 \), then for every complex entry of \( G \), \( G_{i(1 \leq i \leq n)} = [G'] \), it is not the case that \( M, g \models G' \).

- \( M \models G \) iff \( M, f \models G \) where \( f \) is the empty assignment of \( G \) in \( M \).

- \( G \) is satisfiable iff there exists a \( \Sigma \)-structure, \( M \), such that \( M \models G \).

- \( G \) is valid iff for every \( \Sigma \)-structure, \( M \), it holds that \( M \models G \).

- A set of conceptual graphs is interpreted as the conjunction of its elements. Let \( S \) be a set of CGs, \( M, f \models S \) iff \( \forall s \in S(M, f \models s) \).

### 4.2.1.3 Translation to FOL

To provide an alternative and traditional view on the meaning of conceptual graphs, the \( \Phi \)-translation of the previous chapter is extended to negation boxes. A notion of substitution is used to propagate, inside boxes, the translation of \( * \)-markers by variables. In other words, substitutions are the syntactic side of the former assignments.

**4.2.6. Definition. [Substitutions]** Let \( G \) be a conceptual graph, \( c \) be an existential concept node in \( G \), \( m \) be a term and \( t \) be a concept type.

- \( G[c/m] \) is the graph \( G \) in which the marker of every concept node coreferent to \( c \) has been replaced by \( m \).
- \( G[c/(t,m)] \) is the graph \( G \) in which the label of every concept node coreferent to \( c \) has been replaced by \( (t,m) \).

In both cases, the modified concept nodes are eliminated from the domain of the coreference equivalence relation.
As conceptual graphs with variable markers have not been defined, \( G[c/m] \) is a proper conceptual graph iff \( m \in \mathcal{I} \).

4.2.7. Definition. [Extension of \( \Phi \) to CGs] Let \( G \) be a conceptual graph,

1. We first define a translation of concept node markers at the global level of the whole graph. \( \text{term} \) is a function which associates to each concept node in \( G \), a term such that

   \[
   \begin{align*}
   & (a) \forall c \in \{x/x \in C_G \& \text{marker}_G(x) \in \mathcal{T}\}, \text{term}(c) = \text{marker}(c) \quad \text{and} \\
   & (b) \forall c, c' \in \{x/x \in C_G \& \text{marker}_G(x) = \ast\}, \text{term}(c) = \text{term}(c') \in \text{VAR} \iff c \equiv_{\text{co}} c'.
   \end{align*}
   \]

2. We then recursively translate the entries in two steps as follows:

   \[
   \begin{align*}
   & (a) \text{in order to shortcut the quantification of variables in Definition 3.1.16, the chosen variable is substituted to the \ast\text{-marker of every concept node which is "bound by a quantifier occurring in the simple entry of the graph": let } \{c_1, \ldots, c_n\} = \text{Dec}(G), \\
   & \quad G' = G[c_1/\text{term}(c_1)] \ldots [c_n/\text{term}(c_n)] \]
   \]

   \[
   \begin{align*}
   & (b) \text{let } \{x_1, \ldots, x_n\} = \text{term}(\text{Dec}(G)), \\
   & \quad \Phi(G) = \exists x_1 \ldots \exists x_n(\Phi(\text{simple}(G')) \land (\text{complex}(G') \land \neg (\Phi(G''))))
   \end{align*}
   \]

We have presented the translation to FOL as a sidetrack in our tour through conceptual graph fragments, a guideline for anchoring graphical items to their corresponding notions in a well-established framework. Therefore, we will skip both the proof that the translation is in agreement with the former truth definition and the exposition of a reciprocal translation from first-order logic to conceptual graphs that would be required for validating the equivalence of the fragments. Instead, we will focus on the promised proof method that directly manipulates the graphical syntactic items.

4.2.2 Combining tableaux with projections

A keyword of tableau systems is again simplicity: a tableau is model construction by successive decomposition of an input into smaller pieces; hence, the analytical qualification of the method.

These decompositions are guided by the form of the input representations at each stage. Therefore, to any conceptual graph is associated a type depending on the underlying logical operator which is dominating the representation:
4.2.8. **Definition.** [Graph types] Let $G = (\{G_1, \ldots, G_{n(1 \leq n)}\}, co)$ be a conceptual graph. We distinguish three exclusive types of graphs:

1. $G$ is of type $\alpha$ if one of the following conditions holds:
   
   (a) $n = 1$ and $\text{Dec}(G) \neq \emptyset$;
   
   $G$ is a simple graph with existential nodes
   
   (b) $n = 2$ and $\text{simple}(G) \neq G_{\emptyset}$;
   
   $G$ is a conjunction of a non-empty simple graph and the negation of a graph.
   
   (c) $n > 2$;
   
   $G$ is a conjunction of non-empty entries.

2. $G$ is of type $\beta$ if $n = 2$ and $\text{simple}(G) = G_{\emptyset}$ and $\text{Dec}(G') \neq \emptyset$ where $G_2 = \overline{G'}$;
   
   $G$ is the negation of a graph with existential nodes

3. $G$ is of type $\chi$ if one of the following conditions holds:

   (a) $n = 1$ and $\text{Dec}(G) = \emptyset$;
   
   $G$ is a completely instantiated simple graph

   (b) $n = 2$ and $\text{simple}(G) = G_{\emptyset}$ and $\text{Dec}(G') = \emptyset$ where $G_2 = \overline{G'}$;
   
   $G$ is the negation of a completely instantiated simple graph

It is straightforward to verify that the classification defines a partition of the set of all conceptual graphs over a given support.

4.2.9. **Fact.** Any conceptual graph is of one and only one of the three types.

4.2.2.1 **Decomposition rules**

The model construction proceeds by transforming an input graph into a disjunction of conjunctions preserving the satisfiability of the input. The disjunctive form is captured as a tree which represents the disjunction of its branches, while each branch is the conjunction of all nodes occurring on it.

A graph of type $\alpha$ is basically an existentially quantified conjunction of subgraphs. It is decomposed as follows: first, dominating existentially quantified concept nodes, which may be coreferent to other concept nodes in the different subgraphs, are replaced by witnesses (i.e., new constants). Then, the different conjuncts are split along a branch.

By opposition, a graph of type $\beta$ is the negation of an $\alpha$-graph or, in classically equivalent terms, a universally quantified disjunction of subgraphs. Before splitting the disjuncts, the concept nodes corresponding to dominant universal quantifiers are instantiated. Then, the disjunction can be represented by a branching in the tree.
4.2.10. **Definition.** [Tableaux] A tableau is a tree whose nodes are occurrences of CGs. The tableau $\mathcal{T}$ may be extended if one of the following two cases applies.

**α:** an $\alpha$-type graph $\alpha = (\{\alpha_1, ..., \alpha_n \mid 1 \leq n\}, co)$ occurs on the branch $B_H$ from the root to a leaf $H$ in $\mathcal{T}$.

Let $X_\alpha$ be a set of concept nodes in $\text{simple}(\alpha)$ such that $D\text{Dec}(\alpha) \subseteq X_\alpha \subseteq \text{Dec}(\alpha)$.

Let $\Theta_\alpha$ be a substitution which associates a new object name to every concept node in $X_\alpha$; i.e. $\forall x \in X_\alpha$ it holds that (i) $\Theta_\alpha(x) = m \in \mathcal{I}$ and (ii) $m$ does not occur in $\mathcal{T}$ and (iii) $\forall y \in X_\alpha (x \neq y \rightarrow \Theta_\alpha(x) \neq \Theta_\alpha(y))$.

For $1 \leq i \leq n$, we define $\alpha'_i = \text{graph}(\alpha_i)[\Theta_\alpha]$.

We may adjoin successively $\alpha'_1, ..., \alpha'_n$ such that $\alpha'_1$ is the sole successor of $H$ and $\alpha'_{2 \leq i \leq n}$ is the sole successor of $\alpha'_{i-1}$ (if $n = 1$, then $\alpha'_1$ is the sole successor of $H$).

**β:** a $\beta$-type graph $\beta = (\{G_\emptyset, \beta = \overline{\beta'} \}, co)$ occurs on the branch $B_H$ from the root to a leaf $H$ in $\mathcal{T}$. $\beta' = (\{\beta'_1, ..., \beta'_n \mid 1 \leq n\}, co')$ and if $n > 1$, then $\beta'_{i(1 \leq i \leq n)} = \overline{\beta'}_i$.

Let $X_\beta$ be a set of concept nodes in $\text{simple}(\beta')$ such that $D\text{Dec}(\beta') \subseteq X_\beta \subseteq \text{Dec}(\beta')$.

Let $\Theta_\beta$ be a substitution which associates an instantiated label to every concept node in $X_\beta$ such that: $\forall x \in X_\beta$ it holds that (i) $\Theta_\beta(x) = (t, m) \in \mathcal{C} \times \mathcal{I}$ and (ii) if there is a concept node labelled with $(t', m)$ in $\mathcal{T}$, then $t = t'$, else $t \leq_\mathcal{C} \text{type}_\beta(c)$.

$\beta''_i = (\{G_\emptyset, \overline{\beta'}_i \}, co_{\beta'_i})[\Theta_\beta]$ and if $n > 1$, then $\beta''_{i(1 \leq i \leq n)} = \text{graph}(\beta''_i)[\Theta_\beta]$.

We may simultaneously adjoin the graphs $\beta''_1$ to $\beta''_n$ as successors of $H$.

4.2.11. **Example.** To illustrate the decomposition of an $\alpha$-graph, consider the
following tableau extension:

The simple entry of the graph contains two declarations, among which one is dominant. We may instantiate either both concept nodes or only the dominant one. We chose this last option and replace the marker * by an object name which does not occur in the tableau yet: a. The second declaration is left unchanged: the existential node can later be 'handled' by projection.

A branch is interpreted as the conjunction of its nodes. The graph $\alpha$ is an assertion of the conjunction of the two entries $\alpha_1$ and $\alpha_2$. Hence, after the substitution, we may assert both entries on the branch.

To illustrate a $\beta$-application, we can consider the negation of the previous $\alpha$ graph:

The graph $\beta$ asserts a universally quantified disjunction. We may replace these universal quantifiers by any object name (new or already occurring in the tableau, but in the last case, we must ensure the coherence of the typing) and split the branch into the different disjuncts. As in the $\alpha$-case, only dominant declarations need to be instantiated before the splitting.

A branch being a conjunction, it is not satisfiable if it contains the negation of a graph which logically follows from the concatenation of other graphs on the branch; it is then called closed. If all the branches of a tableau are closed then it is not possible to find a model for the represented disjunction of conjunctions. Furthermore, in this last case, as we will prove that the satisfiability of the input graph (i.e., the root of the tree) is preserved by the tableau construction, we can conclude that the input graph is not satisfiable. Hence, a validity proof of a conceptual graph is a closed tableau of the negation of the graph.
4.2.12. DEFINITION. [Proofs] Let $B$ be a branch of a tableau $T$ and $positive(B)$ be the normal form of the concatenation of all simple graphs occurring as node of $B$,

- $B$ closes if it contains a node which is the negation of a simple graph $H$ and there exists a projection from $H$ to $positive(B)$.
- A proof of $G$ is a tableau started with $\boxed{G}$, which has all of its branches closed.

4.2.2.2 Completeness of the calculus

The following proof follows the line of standard tableau completeness proofs. The main differences with usual textual systems reside in the following two features: (i) dominating boolean connectors and quantifiers are here handled in one expansion step and (ii) the search for a contradiction on a branch is performed by projection of simple subgraphs.

4.2.13. LEMMA. If $S$ is a satisfiable finite set of conceptual graphs, then:

**F1:** if an $\alpha$ occurs in $S$ then $S \cup \{\alpha'_1, \ldots, \alpha'_n\}$ is satisfiable.

**F2:** if a $\beta$ occurs in $S$ then at least one of the $n$ sets $S \cup \{\beta''_1\}, \ldots, S \cup \{\beta''_n\}$ is satisfiable.

Proof:

- **F1**, case 1: suppose that $\Theta_\alpha$ is empty. Hence, there is no dominant declaration to replace in $\alpha$. Thus, $S \cup \{\alpha_1, \ldots, \alpha_n\}$ is satisfiable.

  case 2: $\Theta_\alpha$ is not empty. $S \cup \{\alpha\} = S$ is satisfiable, say in a structure $M = (D, F)$. Hence, there is at least one partial assignment $f : X_\alpha \rightarrow D$ such that $M, f \models S \cup \{\alpha\}$ is satisfiable. $\Theta_\alpha$ is a function from $X_\alpha$ to object names that do not occur in $S$, therefore, we can transform $M$ into a model $M' = (D, F')$ such that $M' \models S \cup \{\alpha[\Theta_\alpha]\}$ by defining $F' = f$ on the constants in the codomain of $\Theta_\alpha$ and $F' = F$ on the remaining constants. Now, there are no more dominant declarations in $\alpha[\Theta_\alpha]$, thus, by case 1, we may safely split the conjunction of entries: $S \cup \{\alpha'_1, \ldots, \alpha'_n\}$ is satisfied in $M'$.

- **F2**, case 1: suppose that $\Theta_\beta$ is empty and $S$ is satisfiable. $\beta$ is the negation of a conjunction of graphs that are pairwise not connected by coreferences. From the interpretation of a negation box, it is clear that for at least one of these subgraphs, the conjunction of its negation with $S$ is satisfiable.

  case 2: $\Theta_\beta$ is not empty. $S \cup \{\beta'\}$ is satisfiable, thus it is not the case that there exists a substitution for the declarations of $\beta'$ such that $S \cup \beta'$ is
satisfiable. Hence, for any such substitution, \( \Theta_\beta \), it holds that \( S \cup \{ \beta' [\Theta_\beta] \} \) is satisfiable. \( \beta' [\Theta_\beta] \) is the negation of a conjunction of subgraphs which are not bridged by coreferences, thus, by case 1, it can safely be decomposed.

4.2.14. THEOREM (SOUNDNESS). *If there is a closed tableau started with \( G \), then \( G \) is valid.*

*Proof:* if the root \( G \) is satisfiable then at least one branch is satisfiable (by induction on the number of rules applied and the preservation of satisfiability). Projection is complete for simple graphs (cf. Theorem 3.2.6) and tableau branches (viz. sets of CGs) are interpreted as the conjunction of their nodes, thus a closed branch is unsatisfiable. Hence, one branch of the tableau must be open.

Thus, as the tableau is closed, the root \( G \) must be unsatisfiable and \( G \) is valid. □

It remains to be proved that the tableau construction does not overlook any closing case; i.e, that by applying a systematic procedure that enumerates all possible expansions, if a branch remains open, then it is satisfiable.

4.2.15. DEFINITION. [Complete tableaux] A branch, \( B \), of a tableau, \( T \), is exhausted if for every \( \beta \)-graph in \( B \) and every constructible substitution function, \( \Theta_\beta \), at least one resulting extension \( \beta'' \) occurs in \( B \).

A tableau, \( T \), is called complete if for any branch \( B \) of \( T \), the following holds:

1. if an \( \alpha \) occurs in \( B \) then for at least one choice of the substitution function \( \Theta_\alpha \) all corresponding \( \alpha' \) occur in \( B \), and

2. \( B \) is either closed or exhausted.

Fulfilling the exhaustion may require infinitely many steps. In a systematic procedure, an order on individual markers, e.g. the alphabetical one, must be chosen in order to enable the complete enumeration requested by the exhaustion of \( \beta \) applications. We assume that we have a systematic procedure for constructing a complete tableau, which guarantees that if the process of extending a branch does not terminate, then the resulting infinite branch is exhausted (for example an adaptation of Smullyan’s systematic procedure in [Smu68]).

4.2.16. LEMMA. *In a complete tableau, \( T \), every exhausted open branch, \( B \), is satisfiable.*
4.2. Full classical negation

Proof:

1. We construct a canonical model $M = (D, F)$ from the information conveyed by $\chi$-graphs in $B$.

   (a) $D$ is the set of all object names occurring in $B$.

   (b) The interpretation function $F$ is the identity on $D$ and for every object name $x \in I \setminus D$, $F(x) = \emptyset$.

   (c) By construction of a tableau, an object name can only occur in association to a single concept type. Thus, we can respect the semi-lattice condition, by collecting the type of an object name and propagating this name upward to the interpretation all dominating concept types: for every $d \in D$ and every concept type $t' \in C$, let $t \in C$ be the type of any concept node labelled with $d$ in $B$, it holds that $d \in F(t')$ iff $t \leq C t'$.

   (d) For every relation name $P$ of arity $n$ and every sequence $(d_1, \ldots, d_n) \in D^n$, it holds that $(d_1, \ldots, d_n) \in F(P)$ if and only if there exists in $B$ a (positive) simple conceptual graph in which occurs a relation node whose label is $P'$ such that $P' \leq_R P$ and whose (ordered) arguments are labelled with the object names $(d_1, \ldots, d_n)$.

   It is immediate to verify that $M$ satisfies the hierarchical constraints conveyed by the underlying language signature.

2. We must show that every conceptual graph, $G$, occurring as a node $B$, is satisfiable in the structure $M$. By induction on the degree of $G$.

   (a) If $\text{degree}(G) = 0$, then $G$ is a simple graph.

      i. If $\text{Dec}(G) = \emptyset$, then it is obvious that $G$ is satisfiable in $M$. Indeed, by construction of $M$, every atomic piece of information in $G$ has been included in the structure and if there were two contradictory atomic pieces of information, then the branch would be closed by projection.

      ii. Else, $\text{Dec}(G) \neq \emptyset$ and $G$ is an $\alpha$. Since the tableau is complete, there exists in $B$ a copy of $G$ with all declarations replaced by new object names: say, $\alpha'_1$ obtained from $G$ by applying a substitution $\Theta_\alpha$ (note that tableau rules enable to obtain this instantiated graph in a finite number of successive steps, but for a complete tableau, it is requested to be obtained in a single simultaneous substitution of all declarations). $\alpha'_1$ corresponds to the previous case and thus, it is satisfiable in $M$. Now we can use the function $\Theta_\alpha \cup F$ as an assignment which satisfies $G$ in $M$. 
(b) $\text{degree}(G) > 0$

i. If $G$ is an $\alpha$ (with $n > 1$ as $\text{degree}(G) > 0$), then $\alpha'_1$ to $\alpha'_n$ (obtained with the substitution function $\Theta_\alpha$) occur in $B$. Since $\text{degree}(\alpha'_{i(1 \leq i \leq n)}) < \text{degree}(G)$, then by induction hypothesis, each of these $\alpha'_{i(1 \leq i \leq n)}$s is true under an assignment $f_i$ in $M$. Furthermore, the coreference class of a dominant declaration in $G$ has been instantiated with a single new object name. Hence, each of the $\alpha'_{i(1 \leq i \leq n)}$s is satisfied by $M$ under $f = \bigcup_{1 \leq i \leq n} f_i$. Therefore, $G$ is satisfied by $f \circ \Theta_\alpha$ in $M$.

ii. $G$ is a $\beta$ of the form $[\beta'_1, \beta''_2, ..., \beta''_n]$.

$B$ is exhausted, therefore it holds that for every constructible substitution function $\Theta_\beta$, at least one of the graphs in $\{[\beta'_1|\Theta_\beta]\} \cup \bigcup_{1 \leq i \leq n} [\beta''_i|\Theta_\beta]$ occurs in $B$ and is satisfiable in $M$ (by induction hypothesis and since the degree of all these graphs is less than the degree of $G$). Hence, for every $\Theta_\beta$, $[\beta'_1|\Theta_\beta], [\beta''_2|\Theta_\beta], ..., [\beta''_n|\Theta_\beta]$ is satisfiable in $M$. Thus, it is not the case that there exists a substitution $f : \text{Dec}(G) \rightarrow D$ such that $\forall c \in \text{Dec}(G), f(c) \in F(\text{type}(c))$ and $[\beta'_1, [\beta''_2, ..., [\beta''_n]]_f]$ is satisfiable. Therefore $G$ is satisfiable in $M$.

iii. $G$ is the negation of a simple conceptual graph: $[\beta'_1]$.

Suppose that $G$ is not satisfiable in $M$, then $\beta'_1$ is satisfiable in $M$, viz. there is an embedding of $\beta'_1$ into $M$. Furthermore, by construction, $M$ is the canonical model of the concatenation of all simple conceptual graphs in $B$ (i.e., $\text{positive}(B)$). Hence, by completeness of projection (cf. Theorem 3.2.6), there exists a projection from $\beta'_1$ to $\text{positive}(B)$ and $B$ is closed. But, $B$ is an exhausted open branch, thus $G$ is satisfiable in $M$.

4.2.17. **Theorem (Completeness).** If a conceptual graph $G$ is valid, then there exists a proof of $G$.

**Proof:** A proof of $G$ is a closed tableau started from the negation of $G$: $[\neg G]$. If there is no proof of $G$, then there is an exhausted open branch $B$ in any complete tableau started from $[G]$. By Lemma 4.2.16, $B$ is satisfiable and therefore the root, $[G]$, which is an element of the branch, is satisfiable. Hence, $G$ is not valid.
4.2.3 Related work and further directions

In this brief detour outside our track of efficiently computable graph methods, we have described a general and modular procedure for combining tableau and projection algorithms. Though borrowed from the traditional textual courant of symbolic logic, tableau methods perfectly fit the graphical setting of conceptual graphs: their representation system are trees.

The separation of work between both implied types of calculi, tableaux and homomorphisms, allows their almost independent tuning when practical efficiency is desired. For instance, a well-known technique of tableau implementations consists in delaying the instantiations and delegating the choice to a unification procedure. A free-variable variant of tableaux for conceptual graphs has been proposed in [Ker97]. In this case, projection can be seen as a meta-unification procedure on pieces of information which are not necessarily atomic, but quantified conjunc­tions of atoms.

Peirce’s existential graphs. Other complete calculi for conceptual graph languages equivalent to FOL have been proposed since the early days of conceptual graphs (e.g., [Sow84], [Wer95] or [Ham98]), but their faithfulness to Peirce’s original system of existential graphs have distanced them from possible implementation. Indeed, although some argue for the intuitive characteristics of Peirce’s $\alpha$ and $\beta$ systems (see e.g., [Rob73, Rob92] and [Thi75] for detailed presentation of Peirce original work on FOL calculi), these proof methods are not analytical and rely extremely on human intuition to guess the next proof step. For instance, the following rule is an essential part of the propositional $\alpha$ calculus ([Rob73] p41)

"The rule of insertion. Any graph may be scribed on any oddly enclosed area".

Of course, it is classically valid to derive $\neg(A \land B)$ from $\neg(A)$, but when it comes to use the rule in an automated way, the infinite choice of graphs offered to take the place of $B$ is problematic.

Far from the ideological debate on the ease of learning or using such-and-such type of calculi, we have decided to adopt Peirce’s notion of closed areas to represent negations, while choosing an analytical proof method that a machine can apply without much intuition. After all, the part reserved to cognitive efficiency in this thesis is sufficiently occupied by graphical items.

DRT-Tableaux. In [BE82], it was already advanced that the interaction of tableau reasoning and picture embedding can serve linguistics purposes such as the resolution of some kinds of anaphoric bindings. The existential nature and the sole use of implication and conjunction in discourse representation structures [Kam81] lend themselves to the analytical steps of tableau constructions. [KR96], an early tableau calculus for DRT, has been an important source of inspiration for preliminary versions of this chapter: interlacing projections and tableaux
have been proposed in [Ker96, Ker97] for a language of conceptual graph based on implication.

**Graph rules.** Having implication as sole logical connective between simple conceptual graphs has also been applied in a study of a resolution-like calculus [SM96]. In [Sal97], this method combining resolution steps and adapted projections has been experimentally compared to classical resolution techniques, in order to highlight the advantages of projection for an early detection of dead-ends in the explored proof tree. Comparing the resolution style and the tableau one in the conceptual graph framework has also been our theme in [KS97].

**Intermediate decidable fragments.** Among the infinity of fragments standing between simple graphs and full conceptual graphs, most interesting are the decidable ones for which only a careful application of the (adapted) tableau rules can guarantee to satisfy the sharp bounds laid by theoretical complexity analysis.

One such appealing logic is the guarded fragment of first-order logic (2-EXP-complete satisfiability problem [Grä99]) which expands the tree property to new horizons. A guarded syntax of conceptual graphs has been proposed in [BKM99] and the calculus side would also deserve some attention, witness the subtleties employed in [Niv98] for adapting resolution to guarded FOL. So, the next task for the proponents of a systematic exploration of the conceptual graph landscape lies straight ahead.

For the time being, we propose an alternative route that offers the possibility of closing a loop opened in the previous chapter: reintroducing the tree-like structures of guarded simple graphs into their modal ancestors.

## 4.3 Nested graphs

A keyword in knowledge representation is *modularity*. Imagine how tragic it would be, if to access a web page, the whole content of all internet sites would first need to be downloaded. The world wide web is a typical modular knowledge base in which, among other properties, pieces of information grouped in pages are made available by different authors, pages or even predefined locations in pages are named and can be referred by other pages and in some cases, chosen pieces of information can only be reached by specific sequences of actions. This modular setting is common to almost all kinds of complex representations; e.g., the sectioning units in a book, the division of a cooking receipt into small tasks which can sometimes be accomplished in parallel, a large scale schema with enlargement of specific items, object-oriented programming languages, etc.

When the represented knowledge is partitioned into local subsets, the way navigation from one group to an other is offered, takes a preponderant rôle; after
4.3. Nested graphs

...all, representing information makes sense if it possible to retrieve any piece even in the deepest nested subunit.

Modularity does not only prove salient to representation: we have observed, in Chapter 2, how a switch of interpretation viewpoint, from the global one of first-order logic fragments to the local one of modal logics, can beneficially influence the complexity of reasoning.

In this section, we will combine these representational and computational themes into a framework offering the possibility to localise information conveyed by simple conceptual graphs.

4.3.1 Modularity by nesting

In a single picture, a way to distinguish knowledge levels is to nest pieces of information inside other ones. Here, the term level should not necessary convey a notion of preferability, importance, but rather a geography of information with traced paths between the different pieces. The implicit navigational support associated to nesting is some kind of zooming in procedure which enables to let the focus jump from the level at stake to one of the pieces nested in it.

Nested simple conceptual graphs emerged as an extension of the simple conceptual graph model in [Sow92] but have only been thoroughly studied in recent work [CM97, Sim98, CMS98] on the basis of a better understanding of the underlying simple structures.

4.3.1.1 The traditional conceptual graph approach to nesting

Information is still represented by simple graphs with the difference that an entire graph can be associated to the concept node it describes, can be nested in it.

4.3.1. Example. If I should describe my car, I would not start by exposing the failure of a small non-vital system in one of its components (i.e., the opening of the boot), but rather with the general features of the vehicle (e.g., its age, the engine main characteristics, the type of steering system, etc.):
Building on our understanding of simple graphs, it is most natural to interpret simple graph components of a nested graph in classical first order structures and consider that the notion of nesting corresponds to some relation between the different structures described.

This sketch of semantics already raises many questions about the relation between the simple structures involved in a model for a nested graph. Not only, how they are connected to each other, but also whether they have the same universe, whether nesting is associated to some notion of inheritance, etc. The range of alternatives implied by such questions is familiar to modal predicate logics and the semantics for nested conceptual graphs proposed in the forthcoming sections is inevitably not the unique one, but it will conciliate the chosen syntactic features of nested graphs to the intuitive meaning we will associate to them.

**Individuals and domains**  A nested graph is intended to represent some structured information about a set of objects that may be named with individual markers. Furthermore, the scope of quantification will be allowed to go beyond the boundaries of nested components through coreference links. Hence, it makes sense to associate the same denotation to any individual marker in every substructure where it exists. Concerning this last point, we note that some objects may not be relevant to all substructures: in Example 4.3.1, the boot is not an object that necessarily belongs to the first level of description of my car, whereas it is relevant to the context of the openings. Hence, every substructure will have as domain a subset of the global universe.

**Navigation**  There is a prime syntactical distinction between the notion of nesting in conceptual graphs and the usual notion of modal operator in modal logics: a subgraph is associated to a particular term in a graph, while a modal subformula is usually directly nested in another formula. This notion of term-nesting can be the representation of a relation connecting objects in a substructure to other substructures.

As any individual marker has a fixed denotation in every substructure, should it also have the same connections in every substructure or, on the contrary, should occurrences of an individual marker in distinct subgraphs correspond to objects that have a priori nothing in common except sharing a name? We argue that such a choice should be expressible in the language and hence, that we need to extend the usual syntax of nested conceptual graphs. Let us consider the description of a web-site outline in which two different uses of nesting are interlaced:

**4.3.2. Example.** We have three web-pages: *home*, *page1* and *page2*. The author has designed two buttons called *next* and *home* that are used in the different pages:
- On *home*, $\varphi_{home}$ holds (e.g., $\varphi_{home}$ can be a representation of the content of the page) and the button *next* is a link to *page1*. 
4.3. Nested graphs

- On page1, $\varphi_{\text{page1}}$ holds, next leads to page2 and home to the home-page.
- Finally, on page2, $\varphi_{\text{page2}}$ holds and the button home points to the home-page.

The structure we want to represent by a nested graph has the following outline:

Chein et al [CMS98] proposes two alternative semantics for nested conceptual graphs:

1. In the first one, two occurrences of a term share a priori nothing excepted their name. Therefore, to ensure the capture of the fact that the page pointed by home verifies $\varphi_{\text{home}}$, it is required to nest a copy of the home-page into the concept node [button : home]. Unfortunately, an adequate conceptual graph representation of the site becomes an infinite chain of nesting with no available syntactical shortcut for the repeated pattern:

2. In the second semantics proposed by Chein et al., all occurrences of a term have the same description (i.e., an object is connected to the same substructures in all substructures where it exists). In this case, the representation of the home-link is no more problematic, but now, all occurrences of the
next-button need to be named differently:

![Diagram showing nested subgraphs with different pages and buttons]

The unsatisfactory point in this setting is that it does not represent the fact that both occurrences of the button next correspond to a single object occurring in different pages.

We can attack the problem directly at its source: the missing syntactical element is a way to name a nested subgraph such that latter references to it are enabled. Hybrid logics (e.g., [BT99, ABM99b, Are00]) are tackling a similar problem in modal logics; they introduce nominals, which are terms in a first-order logic manner, that denote worlds in propositional modal languages.

Similarly, we will name every nested subgraph with either a "graph constant" or an indefinite marker representing existential quantification.

4.3.1.2 Exploiting the power of guards in the nested setup

The guarded fragment of simple conceptual graphs (and its sub-fragment of trees) had already a modular flavour: localised pieces of information connected by constrained paths of coreference links. This is not surprising as guarded fragments arose from the study of modal languages. It is interesting to "close a loop" by reimporting the notion of guards into a modal language. Indeed, by including FOL, modal predicate logics inherit undecidable decision problems. However, we will prove that forcing guarded patterns of nesting is a way to highlight a tractable fragment of nested conceptual graphs.

4.3.2 Nested conceptual graphs

A common ground to the different "flat" conceptual graph fragments is a close resemblance of, on one hand, the graph representations and on the other hand, the represented formal structures; actually this homomorphic setting has been
 equally highlighted as the justification of the projection calculus completeness and as a cognitive strength of the graphs.

4.3.2.1 Syntax

Nesting has been intuitively presented as a way of clustering pieces of first-order information into modules of a network. We have argued for the necessity of naming these modules: nominals are introduced in the vocabulary.

4.3.3. Definition. [Extension of a simple graph signature] A nested signature \( \Sigma \) is a pair \( (\sigma, \mathcal{N}) \) where \( \sigma = (\mathcal{I}, (C, \leq_C), (\mathcal{R}, \leq_R), \text{arity}) \) is a signature for simple conceptual graphs\(^2\) and \( \mathcal{N} \) is a non-empty set of nominals with a distinguished element \( N_0 \) and \( \mathcal{I} \cap \mathcal{N} = \emptyset \).

To simplify the reading, we will use lower cases letters for object names in \( \mathcal{I} \) and capital ones for nominals in \( \mathcal{N} \).

In the introductory examples, not all concept nodes were holding a nested subgraph. However, for the sake of simplicity, it is convenient to define a nested conceptual graph as a simple conceptual graph in which a nested conceptual (sub)graph is associated to every concept node. The end-point of the recursion is obtained by taking the empty simple conceptual graph (i.e., the logical constant True) as a nested one.

We adopt a syntactic definition which will later facilitate the translation to the simple graph fragment: instead of using a purely recursive definition, we emphasise the set of simple graph components involved in a nested graph.

4.3.4. Definition. [Syntax of a nested conceptual graph] Given a nested signature \( \Sigma = (\sigma, \mathcal{N}) \), a nested conceptual graph (NCG) over \( \Sigma \) is a finite graph \( G = (D = \{D_0, ..., D_n\}, \text{desc}, \text{nom}, \text{co_desc}, \text{co_trans}) \) where \( 0 \leq n \) and

\(^2\)Definition 3.1.1
1. $D$, the set of descriptions, is a non-empty set of normalised simple con­ceptual graphs over $\sigma$. For $0 \leq i \leq n$, $D_i = (R_{D_i}, C_{D_i}, E_{D_i}, label_{D_i}, co_{D_i})$. The set of concept nodes occurring in $G$ is noted $concept_G$; i.e., $concept_G = \bigcup_{D_i \in D} C_{D_i}$.

2. $\text{desc}$ is a bijection from $concept_G$ to $D \setminus \{D_0\}$; there is a one-to-one correspondence between concept nodes and descriptions (with the exception of the outermost description $D_0$).

3. $\text{nom}$ is a labelling function from $D$ to $\mathcal{N} \cup \{\ast\}$ such that $label(D_0) = \mathcal{N}_0$.

4. $\text{co}_{\text{desc}}$, the coreference relation on descriptions, is an equivalence relation on the set of descriptions labelled with the marker "\ast"; i.e., $\{D_i/D_i \in D$ and $\text{nom}(D_i) = \ast\}$

5. $\text{co}_{\text{trans}}$, the trans-description coreference relation, is an equivalence relation on existential concept nodes in $D$ such that two distinct nodes are coreferent only if they occur in distinct descriptions.

In other words, $\text{desc}$ associates a unique description to every concept node and each description is named with either a nominal or the place holder "\ast".

![Figure 4.5: The underlying tree structure of the NCG in Figure 4.4](image)

The underlying recursive structure of nesting can easily be extracted from $D$ and $\text{desc}$: it is a tree with root $D_0$ such that a description $D_i$ is a direct successor of a description $D_j$ with $0 \leq i, j \leq n$ and $i \neq j$ iff there exists in $D_j$ a concept node $c \in C_{D_j}$ such that $\text{desc}(c) = D_i$. For instance, in Figure 4.5 is represented the underlying tree of the nested graph in Figure 4.4.

It should however be noted that $\text{co}_{\text{trans}}$ and $\text{co}_{\text{desc}}$ are external to this recursive structure. Indeed, two idems occurring in independent branches can be coreferent. This notion of cross-world quantification generalises the usual one in predicate modal logic as, here, the scope of existential quantifiers is not bound by the one of surrounding modalities. For instance, the nested graph in Figure 4.4 represents a
situation in which (i) two objects \( a \) and \( b \) are in relation \( Q \), (ii) in the description of \( a \) there is an object which has the property \( P \) and (iii) the same object has the property \( S \) in the description of \( b \), while this object does not necessarily exist in the world depicted by the root \( D_0 \).

### 4.3.2.2 Nested structures

The intuitive meaning of a graph can be formalised by an embedding into a nested structure. To each description \( D_i \) corresponds a first-order structure, a world \( w_i \), verifying the content of the description (the simple conceptual graph at stake) and for each concept node \( c \) of this description \( D_i \); if \( w_j \) is the world corresponding to the subgraph \( \text{desc}(c) \), then it must hold that \((w_i, [c], w_j)\) is in a ternary accessibility relation \( R_{acc} \) connecting objects in a world (the first two arguments) to other worlds.

#### 4.3.5. DEFINITION. [Nested \( \Sigma \)-structures]

A nested structure over \( \Sigma = (\sigma, \mathcal{N}) \) is a tuple \((W, O, [], S, R_{acc})\) such that:

1. \( W \) is a set of worlds
2. \( O \) is a set of (discourse) objects
3. 
   \[
   [.] : \begin{cases}
   \mathcal{I} \to O \cup \{\emptyset\} \\
   \mathcal{N} \to W \cup \{\emptyset\}
   \end{cases}
   \]
4. \( S \) is a function from \( W \) to the set of \( \sigma \)-structures such that:
   \[
   \forall w \in W, \begin{cases}
   S(w) = (\text{dom}^w, [.]^w) \\
   \text{dom}^w \subseteq O \\
   \forall x \in I, \text{ if } [x] \in \text{dom}^w, \text{ then } [x]^w = [x] \text{ else } [x]^w = \emptyset
   \end{cases}
   \]
5. and \( R_{acc} \) is a ternary relation in \((W \times O \times W)\) such that \((w, o, w') \in R_{acc}\) only if \( o \in \text{dom}^w \).

Nominals, like individual markers, have a fixed denotation. The structure assigned to a world by the function \( S \) is a usual non-nested first-order structures respecting the chosen ordering on concept and relation names as proposed in Definition 3.1.6. Furthermore, the definition forces an individual marker to have the same denotation in all worlds where it exists.

#### 4.3.6. FACT.

An object is in the domain of a world if and only if it is in the denotation of the supremum in the concept type hierarchy at the world:

\[
\forall w \in W, \forall o \in O, o \in \text{dom}^w \iff o \in [c^\top]^w
\]
Proof: Definition 3.1.6 assigns the whole domain of a flat structure to the denotation of the top-concept.

By allowing trans-description coreference links, we have abandoned the recursive syntax of nested graphs and therefore, we define assignments globally on the whole representation:

4.3.7. Definition. [Truth definition of NCGs]
Given a nested signature $\Sigma$, a nested $\Sigma$-structure $M = (W, O, [\cdot], S, R_{acc})$ and a NCG $G = (D, \text{desc}, \text{nom}, \text{co}_{\text{desc}}, \text{co}_{\text{trans}})$ over $\Sigma$,
- an assignment is a function $[\cdot]_a$ extending $[\cdot]$ from names to graph items (concept nodes and descriptions) and respecting coreferences:

$$[\cdot]_a : D \rightarrow W$$

$$\text{concept}_G \rightarrow O$$

such that

$$\forall x \in D, \text{nom}(x) \in \mathcal{N} \Rightarrow [x]_a = [\text{nom}(x)]$$
$$\forall x, y \in D, (x, y) \in \text{co}_{\text{desc}} \Rightarrow [x]_a = [y]_a$$
$$\forall c \in \text{concept}_G, \text{marker}(c) \in \mathcal{I} \Rightarrow [c]_a = [\text{marker}(c)]$$

$$\forall c, c' \in \text{concept}_G, (c, c') \in \text{co}_{\text{trans}} \Rightarrow [c]_a = [c']_a$$

- $M \models_{\text{NCG}} G$ iff there exists an assignment $[\cdot]_a$ s.t. $(M, [N_0], [\cdot]_a) \models_{\text{NCG}} D_0$

- $(M, w, [\cdot]_a) \models_{\text{NCG}} D_i$ iff

$$\forall c \in C_{D_i}, [c]_a \in \text{dom} w$$
$$\forall c \in C_{D_i}, (w, [c]_a, [\text{desc}(c)]_a) \in R_{acc}$$
$$\forall c \in C_{D_i}, (M, [\text{desc}(c)]_a, [\cdot]_a) \models_{\text{NCG}} \text{desc}(c)$$

$(S(w), [\cdot]_a) \models_{\text{SCG}} D_i$

- $G \subseteq_{\text{NCG}} H$ iff for every $\Sigma$-structure $M$, if $M \models_{\text{NCG}} G$, then $M \models_{\text{NCG}} H$

We now have a formal grasp on the meaning of nested representations, but how complex is reasoning in the nested framework?

4.3.3 Complexity and Guards

We have defined representations and structures that resemble each other and it should be quite natural to expand the strategy applied to simple graphs: the definition of a projection calculus which simulates the embedding of a nested graph into the canonical model of another one. Chein et al. [CM97, CMS98] define a recursive projection-procedure based on the calculus for simple graphs and they handle the alternative semantics for their nested language by defining alternative
canonical models (via different forms of normalised nested graphs). However a such line of thinking does not provide an immediate clue on the complexity of the recursive method (only a lower NP bound due to the matching of two simple graphs).

We here choose a different strategy: we skip the definition of a calculus and translate the nested setup into a fragment of simple conceptual graphs. The completeness of such a translation would have two corollaries: a measure of the difficulty of reasoning in the nested fragment and a direct exploitation of the tractability of guarded simple graphs (cf. Chapter 3.3).

### 4.3.3.1 From nested graphs to simple ones

A first step concerns a way of encoding the notions of world-partition and local-substructure into a usual first-order structure.

By associating a local-substructure to every world, we have let the properties of each individual (i.e., its concept type and the relations that link it to other individuals of the world at stake) be relative to the local notion of world. This information can be captured by an extra argument to every predicate. Furthermore, from this encoding, we can also derive the domain of a given world: an individual \( o \) belongs to the domain of a world \( w \) if and only if \( w \) and \( o \) occur together in the interpretation of the super-concept \( c^T \). The transitions from objects in worlds to other worlds were captured by the accessibility relation \( R_{acc} \); It will prove convenient for guarded representations to include this information in the denotation of other predicates.

#### 4.3.8. Definition. [Derived signature]
Given a nested signature \( \Sigma = ((\mathcal{I}, (C, \leq_C), (R, \leq_R), arity), N) \), its derived signature is a signature for simple conceptual graphs \( \sigma = (\mathcal{I} \cup N, \{c^\sigma\}, (R^\sigma, \leq_{R^\sigma}), arity^\sigma) \) such that:

1. \( c^\sigma \) is the unique concept type
2. \( \forall r \in R^\sigma, \exists k \in \mathbb{N}^+ \) such that \( arity^\sigma(r) = 2k + 1 \)
3. \( R_3^\sigma = C \cup R_1 \) and \( \leq_{R_3^\sigma} \) preserves both orders \( \leq_C \) and \( \leq_{R_1} \).
4. \( \forall k > 1, (R_{2k+1}^\sigma, \leq_{R_{2k+1}^\sigma}) = (R_k, \leq_{R_k}) \)

How the arity-change is employed becomes clear in the following transformations:

#### 4.3.9. Definition. [\( \lambda-\mu \) translations between nested structures and flat ones]
Given a nested signature \( \Sigma \) with derived signature \( \sigma \), let \( c^T \) be the supremum in the concept type hierarchy of \( \Sigma \),

- For a nested \( \Sigma \)-structure \( M = (W, O, [\cdot]_M, S, R_{acc}) \),
  \( \lambda(M) \) is a flat \( \sigma \)-structure \( (X, [\cdot]_{\lambda(M)}) \) such that
1. relevant worlds:
   \[ W' = [\mathcal{M}]_M \cup \{w \in W/ \exists x \in W, \exists o \in O, \langle x, o, w \rangle \in R_{acc}\} \]
relevant objects:
   \[ O' = [\mathcal{I}]_M \cup \{o \in O/ \exists x, y \in W', \langle x, o, y \rangle \in R_{acc}\} \]
   \[ X = W' \cup O' \]

2. \[ \forall x \in \mathcal{I} \cup \mathcal{N}, \]
   \[ [x]_{\lambda(M)} = [x]_M \]

3. \[ \forall P \in \mathcal{R} \cup \mathcal{C} \text{ such that } \text{arity}(P) = k, \]
   \[ [P]_{\lambda(M)} = \begin{cases} \langle w, o_1, v_1, \ldots, o_k, v_k \rangle/ & w \in W' \\
& \& \langle o_1, \ldots, o_k \rangle \in [P]_M^w \\
& \& \forall 1 \leq i \leq k, \langle w, o_i, v_i \rangle \in R_{acc} \end{cases} \]
   - For a flat \( \sigma \)-structure \( M' = (X, [\mathcal{I}]_M') \),
     \( \mu(M') \) is a nested \( \Sigma \)-structure \( (W, O, [\cdot]_\mu(M'), S, R_{acc}) \) such that
     1. \[ R_{acc} = \{ \langle w, o, v \rangle/ \langle w, o, v \rangle \in [c^T]_M' \] and \( v \) is uniformly accessible from \( o \) in \( w \) \}
       where \( v \) is uniformly accessible from \( o \) in \( w \)
       iff \[ \forall P \in \mathcal{R} \cup \mathcal{C} \text{ such that } \text{arity}(P) = k \]
       and \[ \forall \langle w, o_1, v_1, \ldots, o_k, v_k \rangle \in [P]_M' \]
       if \( o_i = o \) with \( 1 \leq i \leq k \), then \( \langle w, o_1, v_1, \ldots, o_k, v_k \rangle[v_i/v] \in [P]_M' \]
     2. \[ W = [\mathcal{M}]_M' \cup \{w \in X/ \exists x, y \in X \text{ and } \langle x, y, w \rangle \in R_{acc}\} \]
     3. \[ \forall w \in W, \]
       (a) \[ \text{dom}^w = \{o \in X/ \exists x \in W \text{ and } \langle w, o, x \rangle \in R_{acc}\} \]
       (b) \[ \forall x \in \mathcal{I}, [x]_{\mu(M')}^w = \begin{cases} [x]_M' \text{ if } [x]_M' \in \text{dom}^w \\
\& \& \text{otherwise} \end{cases} \]
       (c) \[ \forall P \in \mathcal{R} \cup \mathcal{C} \text{ such that } \text{arity}(P) = k, \]
       \[ [P]_{\mu(M')}^w = \{ \langle o_1, \ldots, o_k \rangle/ \langle w, o_1, v_1, \ldots, o_k, v_k \rangle \in [P]_M' \] and \( 1 \leq i \leq k, \langle w, o_i, v_i \rangle \in R_{acc} \} \]
       (d) \[ S(w) = (\text{dom}^w, [\cdot]^w) \]
     4. \[ O = [\mathcal{I}]_M' \cup \bigcup_{w \in W} \text{dom}^w \]
     5. \[ \forall x \in \mathcal{I} \cup \mathcal{N}, [x]_{\mu(M')} = [x]_M' \]

We note that both structure translations do not preserve those pieces of information which cannot play a rôle in the satisfaction of a nested graph. For

\[^3\]By convention, the arity of a concept type in \( \mathcal{C} \) is 1.
instance, a world which is neither in the denotation of nominals nor in the accessibility relation is eliminated by $\lambda$. Also, world connections are conveyed by the accessibility relation in a nested structure, while the information is duplicated in the arguments of relations in a flat structure and $\mu$ eliminates some incomplete patterns; e.g., suppose that a flat structure $M'$ is partially described by (i) $\{\langle w, a, y \rangle \} = [P]_{M'}$ and $\{\langle w, a, x \rangle \} = [Q]_{M'}$. Neither $x$ nor $y$ is uniformly accessible from $a$ in $w$, hence, (i) is not preserved in $\mu(M')$.

4.3.10. DEFINITION. [Interpretation of NCGs into flat structures] Given a nested signature $\Sigma$ with derived signature $\sigma$, a $\sigma$-structure $M = (O, [\cdot])$ and a NCG $G = (D, desc, nom, co_{desc}, co_{trans})$ over $\Sigma$,

- an assignment is a function from nodes to objects which coincides with the denotation of constants $[\cdot]_a : D \cup concept_G \rightarrow O$ such that

$$\forall x, y \in D \text{ and } \forall c, c' \in concept_G, \begin{cases} nom(x) \in N \Rightarrow [x]_a = [nom(x)] \\ (x, y) \in co_{desc} \Rightarrow [x]_a = [y]_a \\ marker(c) \in I \Rightarrow [c]_a = [marker(c)] \\ (c, c') \in co_{trans} \Rightarrow [c]_a = [c']_a \end{cases}$$

- $M \models_{NCG_{flat}} G$ iff $\exists [\cdot]_a$ assignment s.t. $(M, [N_0], [\cdot]_a) \models_{NCG_{flat}} D_0$

- $(M, w, [\cdot]_a) \models_{NCG_{flat}} D_i$ iff

$$\forall c \in C_{D_i}, \forall r \in R_{D_i}, \begin{cases} \langle w, [c]_a, [desc(c)]_a \rangle \in [type(c)] \\ \langle w, [c]_a, [desc(c_1)]_a, \ldots, [c_k]_a, [desc(c_k)]_a \rangle \in [label_{D_i}(r)] \\ (M, [desc(c)]_a, [\cdot]_a) \models_{NCG_{flat}} desc(c) \end{cases}$$

where $k$ is the arity of the relation symbol labelling $r$ and $c_{j,1 \leq j \leq k}$ is the $j$th concept node neighbour of $r$ in $D_i$.

- $G \models_{NCG_{flat}} H$ iff for every $\sigma$-structure $M$, $M \models_{NCG_{flat}} G$ implies that $M \models_{NCG_{flat}} H$

In Theorem 4.3.12, we will prove that the model transformation $\lambda$ and $\mu$ preserve the satisfaction of nested graphs. For the time being, we will go directly to the point of this section: the translation of a nested graph into a simple one.

4.3.11. DEFINITION. [K translation] Given a nested signature $\Sigma$ with derived signature $\sigma$ and a NCG $G$ over $\Sigma$, $\kappa(G) = (R, C, E, label, co)$ is a simple conceptual graph over $\sigma$ defined from $G = (D, desc, nom, co_{desc}, co_{trans})$ as follows:

1. $\forall D_i = (R_{D_i}, C_{D_i}, E_{D_i}, label_{D_i}, co_{D_i}) \in D$, ...
Chapter 4. Richer pictures

Figure 4.6: The $\kappa$-translation of the NCG in Figure 4.4

(a) $\kappa_{\text{nom}}(D_i)$ is a concept node $c^\sigma: \text{nom}(D_i)$
i.e., to each description corresponds a concept node referring to thename of the described world

$\kappa_{\text{marker}}(c)$ is a concept node $c^\sigma: \text{marker}(c)$

(b) $\forall c \in C_{D_i}$,

$\kappa_{\text{type}}(c)$ is a relation node $\{\text{type}(c)\}$

$\kappa_{\text{edge}}(c) = \{\text{(type}(c), \kappa_{\text{nom}}(D_i), 1)\}$

$\kappa_{\text{edge}}(c) = \{\text{(type}(c), \kappa_{\text{marker}}(c), 2)\}$

$\kappa_{\text{edge}}(c) = \{\text{(type}(c), \kappa_{\text{nom}}(desc(c)), 3)\}$
i.e., a concept node is transformed into a relation node with as argu­ments the description in which it occurs, its marker and the description it is pointing to.

(c) $\forall r \in R_{D_i}$, $\kappa(r) = (r, \kappa_{\text{nom}}(D_i), 1)$ and $\forall e = (r, c, k) \in E_{D_i}$,$\kappa(e) = \{(r, \kappa_{\text{marker}}(c), 2k), (r, \kappa_{\text{nom}}(desc(c)), 2k + 1)\}$
i.e., arities of relation are changed according to the transition from a
nested signature to its derived one.

2.

$R = \bigcup_{D_i \in D} R_{D_i} \cup \bigcup_{c \in \text{concept}_G} \kappa_{\text{type}}(c)$

$C = \bigcup_{D_i \in D} \kappa_{\text{nom}}(D_i) \cup \bigcup_{c \in \text{concept}_G} \kappa_{\text{marker}}(c)$

$E = \bigcup_{D_i \in D} \kappa(E_{D_i}) \cup \bigcup_{D_i \in D} \kappa(R_{D_i}) \cup \bigcup_{c \in \text{concept}_G} \kappa_{\text{edge}}(c)$

$\text{label} = \bigcup_{D_i \in D} \text{label}_{D_i}(R_{D_i}) \cup \bigcup_{c \in \text{concept}_G} \text{label}(\kappa_{\text{marker}}(c)) \cup \bigcup_{c \in \text{concept}_G} \text{label}(\kappa_{\text{type}}(c))$
4.3. Nested graphs

\[
\text{type}(c):\text{marker}(c), \text{nom(desc}(c)):\text{desc}(c) = c \in D_i
\]

In Figure 4.6, the typing of concept nodes has not been represented in the \(k\)-translation. Indeed, the unique concept type does not convey any useful information; it is just a necessary component of the simple conceptual graph syntax.

We need to prove that the proposed translations are meaning preserving:

4.3.12. **Theorem (Completeness of the Translation).** Given a nested signature \(\Sigma\) with derived signature \(\sigma\) and two nested conceptual graphs \(G\) and \(H\) over \(\Sigma\),

\[
G \sqsubseteq_{\text{NCG}} H \iff G \sqsubseteq_{\text{NCG,flat}} H \iff \kappa(G) \sqsubseteq_{\text{SCG}} \kappa(H)
\]

4.3.13. **Lemma.** Given a nested signature \(\Sigma\) with derived signature \(\sigma\) and a nested conceptual graphs \(G\) over \(\Sigma\),

1. \(\forall M\) nested \(\Sigma\)-structure, \(M \models_{\text{NCG}} G \iff \lambda(M) \models_{\text{NCG,flat}} G\)

2. \(\forall M\) flat \(\sigma\)-structure, \(M \models_{\text{NCG,flat}} G \iff \mu(M) \models_{\text{NCG}} G\)

3. \(\forall M\) flat \(\sigma\)-structure, \(M \models_{\text{NCG,flat}} G \iff M \models_{\text{SCG}} \kappa(G)\)

Proof of Lemma 4.3.13(1): let \(M = (W, O, [\cdot]_M, S, R_{acc}), \lambda(M) = (X, [\cdot]_{\lambda(M)})\),
We skip the proof of the reciprocal and the one of Lemma 4.3.13(2) which are very similar checks that enough information is conveyed by the model translations. Proof of Lemma 4.3.13(3): by induction on the structure of $G$.

Let $[\_]_f = [\kappa_{nom}(D)]_a \cup [\kappa_{marker}(concept_G)]_a$

1. $(M, [D_i]_a, [\_]) \models_{NCGflat} c \in D_i$

   with $c = \{type(c) : marker(c), nom(desc(c)) : desc(c)\}$

   iff $([D_i]_a, [\_], [desc(c)]_a) \in [type(c)]$ and $(M, [desc(c)]_a, [\_]) \models_{NCGflat} desc(c)$

   iff $([\kappa_{nom}(D_i)]_f, [\kappa_{marker}(c)]_f, [\kappa_{nom}(desc(c))]_f) \in [type(c)]$

   and $(M, [desc(c)]_a, [\_]) \models_{NCGflat} desc(c)$

   iff $(M, [\_]) \models_{SCG} \kappa(graph(c))$

   where $graph(c)$ is the SCG composed of the single node $c$

2. $(M, [D_i]_a, [\_]) \models_{NCGflat} r(c_1, ..., c_k) \in D_i$

   iff $([D_i]_a, [\_], [desc(c_i)]_a, ..., [\_], [desc(c_k)]_a) \in [label_{D_i}(r)]$

   iff $(M, [\_]) \models_{SCG} \kappa(r(c_1, ..., c_k))$

3. assignments take into account coreferences.
4.3. Nested graphs

It is now straightforward to prove the completeness of the translations.

Proof of Theorem 4.3.12:

\[ \forall \Sigma - \text{structure } M, \quad M \models_{\text{NCG}} G \Rightarrow M \models_{\text{NCG}} H \]
\[ \Downarrow \text{Lemma 4.3.13(1)} \]
\[ \lambda(M) \models_{\text{NCGflat}} G \Rightarrow \lambda(M) \models_{\text{NCGflat}} H \]

\[ \forall \sigma - \text{structure } M', \quad M' \models_{\text{NCGflat}} G \Rightarrow M' \models_{\text{NCGflat}} H \]
\[ \Downarrow \text{Lemma 4.3.13(2)} \]
\[ \mu(M') \models_{\text{NCG}} G \Rightarrow \mu(M') \models_{\text{NCG}} H \]

Hence, \( G \subseteq_{\text{NCG}} H \) iff \( G \subseteq_{\text{NCGflat}} H \)

\[ \forall \sigma - \text{structure } M', \quad M' \models_{\text{NCGflat}} G \Rightarrow M' \models_{\text{NCGflat}} H \]
\[ \Downarrow \text{Lemma 4.3.13(3)} \]
\[ M' \models_{\text{SCG}} \kappa(G) \Rightarrow M' \models_{\text{SCG}} \kappa(H) \]

Hence, \( G \subseteq_{\text{NCGflat}} H \) iff \( \kappa(G) \subseteq_{\text{SCG}} \kappa(H) \)

4.3.3.2 Complexity of reasoning in nested graphs

Through the previous correspondences, we can directly harvest some results on the complexity of our benchmark problems in the nested framework.

4.3.14. COROLLARY. • No nested graph is valid.
• A nested graph is always satisfiable.
• \( G \subseteq_{\text{NCG}} H \) is NP-complete, for \( G \) and \( H \) nested conceptual graphs over a common nested-signature.

Proof: It comes to no surprise that validity and satisfiability are not informative notions for positive graphs.
• By analogy to flat structures having an empty universe, a nested structures can have an empty set of worlds, hence providing no world to interpret the description \( D_0 \) and thus no possible assignment.
• A nested graph \( G \) is always satisfied by the \( \mu \)-translation of the canonical model of \( \kappa(G) \) (cf. Fact 3.2.4).
• By Theorem 4.3.12, subsumption of nested graphs is in NP. Indeed, the \( \kappa \)-translation is polynomial: for a nested graph \( G \) with \( c \) concept nodes, \( r \) relation nodes and \( e \) edges, \( \kappa(G) \) is a graph with \( (3c + 1) \) concept nodes, \( (r + c) \) relation nodes and at most \( (3c + 3e) \) edges. The NP lower bound is the one of subsumption in simple graphs as we can encode any simple graph \( G \) as a nested graph \( \kappa^{-1}(G) \)
with a single non-empty description, $D_0$, and with each concept node described by an instance of the empty graph labelled with $N_0$. It is obvious that $G \subseteq_{SCG} H$ iff $\kappa^{-1}(G) \subseteq_{NCG} \kappa^{-1}(H)$ as any concept node of the premiss introduces a required link to the world $[N_0]$. ■

Despite the cognitive impact of a modular layout of represented information, the expressive power of the nested fragment is the same as the one of the simple graph fragment. For tractability reasons, we can turn ourselves to guarded quantification.

4.3.15. COROLLARY. Given a nested signature $\Sigma$ with derived signature $\sigma$ and two nested conceptual graphs $G$ and $H$ over $\Sigma$, $G \subseteq_{NCG} H$ is decidable in polynomial time if $\kappa(H)$ is a guarded simple graph.

Proof: Immediate consequence of Theorem 4.3.12 and Theorem 3.3.7. ■

4.3.4 Concluding remarks

In this section, we have presented how some modular knowledge can be represented by a language of nested graphs. To overcome a syntactical lack in the traditional setting of nested conceptual graph, we have introduced nominals. As consequence, the different kinds of knowledge found in the literature under alternative semantics ([CM97, Sim98, CMS98]) can now be expressed in a single framework. Furthermore, by translation to the simple conceptual graph fragment, we have prove the low, though untractable, complexity of reasoning in the nested framework and we have isolated a fragment of nested graphs for which subsumption is tractable.

4.3.4.1 Related work

Peirce’s gamma-graphs. Under the name of gamma-system (see e.g., [Rob73, Thi75]), Peirce also studied a modal version of its propositional graphs, but his work remained rather informal and unfinished and, so far, the gamma-system has not yet proved any computational nor cognitive appeals compared to textual propositional logic.

Hybrid and Description logics. Hybrid logics (see for instance Areces’ recent thesis [Are00]) and their use of nominals to name worlds in propositional modal logics have been a source of inspiration for the proposed setting of nested graphs. It should be noted that most complexity studies in hybrid logics have focussed on rich fragments for which satisfiability is often difficult or even undecidable.
Description logics are hybrid logics with a different syntax and a high concern for low complexity reasoning tasks. Most relevant to nested conceptual graphs is the fragment $\mathcal{ELTRO}$, a propositional modal logic with only diamonds, conjunctions and nominal-constants for which Baader et al. [BMT99] propose a graph representation of formulae and a tractable calculus based on graph homomorphisms.

Context logics. Among predicate modal logics, context logics (e.g., [Buv96, MB97, Buv98]) have in common with nested conceptual graphs that modalities are associated to terms. Syntactically, a description $\phi$ attached to a term $t$ is represented by a modal formula $\text{ist}(t, \phi)$ (which reads “from the present context, $\phi$ is true in the context of $t$”). In [Ker99b], a link between nested graphs and the context logic of Buvac is further explored and a calculus combining modal-tableaux and nested projections is proposed for an extension of the language containing negation.

4.3.4.2 Further work.

Guarded nested graphs have been defined as nested graphs that translate into guarded simple ones. It remains to capture on nested representations the constraints that force the guarded property but a formal presentation would require some extensions of the syntax; for instance, crazed graphs (Definition 3.1.13) would need to be able to take part of nested graphs whereas we have deliberately simplify the setting by only allowing descriptions in normal form. An obvious necessary condition is that every description in a nested graph must be a guarded simple graph. Studying the guarded structure of coreference between subgraphs is left for further work.

Secondly, the extension of the language to some restricted forms of negation, such as the limited atomic negation of Chapter 4.1, would also be interesting. Indeed, from description logics and modal propositional logics, we know the tractability of reasoning in some modal fragments including negation. How would guards interact with negations in the modal predicate framework?

Finally, other forms of nesting would also be worth studying. For instance, instead of nesting graphs inside concept nodes, we could have adopted a more usual modal syntax by nesting a graph on the sheet of assertion of another graph (in the style of negated zones in Chapter 4.2). How useful could graph methods be in fragments of usual modal predicate logics?
Hybrid and Description Logics. Hybrid logics (under its manifestations, such as KSL, and their use of relations to define contexts) and propositional modal logics have been a source of inspiration for the proposed use of hybrid logics. It should be noted that most complexity results in hybrid logics have been on logics that are propositional, for which understanding is given in the context of complexity and tractability.