Anti-Prenexing and Prenexing for Modal Logics (Extended Version)

Cláudia Nalon[1] and Clare Dixon[2]

[1] Departamento de Ciência da Computação, Universidade de Brasília
Caixa Postal 4466 – CEP:70.910-090 – Brasília – DF – Brazil
nalon@unb.br

[2] Department of Computer Science, University of Liverpool
Liverpool, L69 7ZF – United Kingdom
C.Dixon@csc.liv.ac.uk

Abstract. Efficient proof methods for proving properties specified by means of normal modal logics are highly desirable, as such logical systems have been widely used in computer science to represent complex situations. Resolution-based methods are often designed to deal with formulae in a normal form and the efficiency of the method (also) relies on how efficient (in the sense of producing fewer and/or shorter clauses) the translation procedure is. We present a normal form for normal modal logics and show how the use of simplification, for specific normal logics, together with anti-prenexing and prenexing techniques help us to produce better sets of clauses.

1 Introduction

Beliefs, knowledge, intentions, desires, and obligations of agents as well as the behaviour of these (and possibly other) aspects over time are often used to describe complex situations in computer science. This is the case, for instance, in the specification of distributed [3] and multi-agent systems [9]. Normal modal logics are often chosen to model and reason about these situations. Given a logical specification, an automated tool such as, for instance, a theorem prover, can then be used for verifying properties of the system. However, in order to model the different aspects of a complex, particular situation, it may be necessary to combine different logical languages. When the combination is given by the *fusion* of logical systems, that is, when the components are independently axiomatisable, proofs can be obtained by combining the provers for each language. Combining those provers may require special care such that all relevant information is correctly handled and exchanged between the different tools. Also, this may require the use of tools which are based on different implementations (e.g. different input languages) or, worse, on different approaches (e.g. partially based on translation to first-order language $\times$ partially based on the modal language, resolution $\times$ tableau, etc), making this task harder.

We are currently investigating a uniform approach which deals with theorem proving for a variety of *propositional normal modal logics*, that is, logics in which the schema $\Box(\varphi \Rightarrow \psi) \Rightarrow (\Box\varphi \Rightarrow \Box\psi)$ (the axiom **K**), where $\Box$ is the modality for *necessity* and φ and ψ are well-formed formulae, is valid. We are interested in multi-modal normal logics based on the following axioms

$$\begin{array}{rl} \mathbf{K}: & \Box(\varphi \Rightarrow \psi) \Rightarrow (\Box\varphi \Rightarrow \Box\psi) \\ \mathbf{T}: & \Box\varphi \Rightarrow \varphi \\ \mathbf{D}: & \Box\varphi \Rightarrow \Diamond\varphi \\ \mathbf{4}: & \Box\varphi \Rightarrow \Box\Box\varphi \\ \mathbf{5}: & \Diamond\varphi \Rightarrow \Box\Diamond\varphi \\ \mathbf{B}: & \Diamond\Box\varphi \Rightarrow \varphi \end{array}$$

where $\Diamond$ is the modality for *possibility*. We formally introduce the weakest of these logics, $\mathsf{K}_{(n)}$, in Section 2. The proof method for each logic is resolution-based and our approach is clausal: in order to prove that a formula, φ, is valid, we first transform its negation, $\neg\varphi$, into a clausal normal form. Here we do not focus on the proof method, but on the properties that a normal form for these logics should have in order to achieve efficiency. We discuss briefly some aspects that should be considered when designing a normal form for a *family* of logics.

Firstly, in the propositional case there is only one resolution rule to apply to the set of clauses, whilst we often need several rules for dealing with modal logics. This happens because the semantics of modal logics are relative to a set of worlds, so we often need to perform reasoning tasks which are not local (to the actual world). Also, if we consider multi-agents contexts, the resolution rules must consider the different contexts (relative to each agent) in which the reasoning applies. Separating these contexts may facilitate the reasoning task. Thus, it should be taken into consideration when designing a normal form for these logics.

Also, we should think of strategies that could be used to reduce the proof search. Our work is based on that of [1], but the normal form differs slightly (where l_i are literals, i.e., propositions or their negations): clauses are separated into literal clauses (disjunctions of literals), positive modal clauses (an implication as $l_1 \Rightarrow \Box l_2$), and negative modal clauses (an implication as $l_1 \Rightarrow \neg\Box l_2$). This further separation potentially allows a better design of strategies to guide the search for a proof when applying the resolution method. For instance, complete strategies for (purely) propositional logic could be used when the parents are literal clauses. The set of support strategy could also be used when the parents are modal clauses, taking, for instance, the positive modal clauses in the usable set and the negative modal clauses in the set of support (or vice-versa).

The new normal form is given in Section 3. Transformation into clausal form is carried out by performing classical style rewriting, simplification, and renaming [8], a technique which may avoid combinatorial explosion on the size of the formula by replacing complex subformulae by new symbols, whose meaning are linked to the formula that they are replacing.

Efficient translation is crucial for practical use of the resolution method. By *efficient* we mean that the translation method produces fewer or shorter clauses. In first-order logic, it has been shown [2] that the transformation of a given problem into anti-prenex normal form (i.e. when quantifiers are moved inwards a formula) results in a better set of clauses. However, to the best of our knowledge, there has not been a similar investigation for modal logics. We present an algorithm for anti-prenexing in Section 4. Experimentally, anti-prenexing together with simplification, given in Section 5, followed by transformation into the normal form performs better than both the transformation preceded by anti-prenexing and transformation alone.

We introduce the prenex normal form in Section 6 and show how this can also be used to reduce the nesting of modal operators *after* anti-prenexing. Simplification for specific logics is used in both steps, anti-prenexing and prenexing. Preliminary results show that the set of clauses is usually smaller than that obtained by translation into the normal form alone.

Experimental results are given in Section 7. We provide concluding remarks in Section 8.

2 The Basic Normal Logic

The basic normal modal logic that we present here is known as $\mathsf{K}_{(n)}$. This is the weakest of the normal modal systems, where only the distribution axiom (the axiom **K**) holds. There is no restriction on the accessibility relation over worlds. As the subscript in the name of the logic indicates, we consider the multi-agent version, given by the fusion of several copies of $\mathsf{K}_{(1)}$, one for each agent.

2.1 Syntax

Formulae are constructed from a denumerable set of *propositional symbols*, $\mathcal{P} = \{p, q, p', q', p_1, q_1, \ldots\}$. The finite set of agents is defined as $\mathcal{A} = \{1, \ldots, n\}$. In addition to the standard propositional connectives $(\neg, \vee, \wedge, \Rightarrow)$, we introduce a set of unary modal operators $\boxed{1}, \ldots, \boxed{n}$, where $\boxed{i}\varphi$ is read as "agent i considers φ necessary". When $n = 1$, we may omit the index, that is, $\Box\varphi = \boxed{1}\varphi$. We do not define the operator $\Diamond$: the fact that an "agent i considers φ possible" is expressed by $\neg\boxed{i}\neg\varphi$. The language of $\mathsf{K}_{(n)}$ is defined as follows:

Definition 1. *The* ***set of well-formed formulae****,* $\mathsf{WFF}_{\mathsf{K}_{(n)}}$*:*

- *the propositional symbols are in* $\mathsf{WFF}_{\mathsf{K}_{(n)}}$*;*
- **true** *and* **false** *are in* $\mathsf{WFF}_{\mathsf{K}_{(n)}}$*;*
- *if* φ *and* ψ *are in* $\mathsf{WFF}_{\mathsf{K}_{(n)}}$*, then so are* $\neg\varphi$*,* $(\varphi \wedge \psi)$*,* $(\varphi \vee \psi)$*,* $(\varphi \Rightarrow \psi)$*, and* $\boxed{i}\varphi$ $(\forall i \in \mathcal{A})$*.*

The following definitions will also be used later.

Definition 2. *A* ***literal*** *is either a proposition or its negation.*

Definition 3. *A* ***modal literal*** *is either* $\boxed{i}l$ *or* $\neg\boxed{i}l$*, where* l *is a literal and* i *is in the set of agents,* $\mathcal{A} = \{1, \ldots, n\}$*.*

Definition 4. *A formula χ is **disjunctive** if, and only if, is of the form $(\varphi \Rightarrow \psi)$, $(\varphi \vee \psi)$ or $\neg(\varphi \wedge \psi)$. Otherwise, χ is said to be **conjunctive**.*

Polarity of a formula is defined as usual: if a formula is inside the scope of an even (including zero) number of negation symbols, the formula is said to be of *positive* polarity; otherwise, it is of *negative* polarity.

2.2 Semantics

Semantics of $\mathsf{K}_{(n)}$ is given, as usual, in terms of a Kripke structure.

Definition 5. *A **Kripke structure** M for n agents over $\mathcal{P}$ is a tuple $M = \langle \mathcal{S}, \pi, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$, where $\mathcal{S}$ is a set of possible* worlds *(or* states*) with a distinguished world s_0 ; the function $\pi(s) : \mathcal{P} \rightarrow \{true, false\}$, $s \in \mathcal{S}$, is an interpretation that associates with each state in $\mathcal{S}$ a truth assignment to propositions; and $\mathcal{R}_i$ is a binary relation on $\mathcal{S}$.*

The binary relation $\mathcal{R}_i$ is intended to capture the possibility relation according to agent i. So, a pair (s, t) is in $\mathcal{R}_i$ if agent i considers world t possible, given her information in world s. In $\mathsf{K}_{(n)}$, the relations are any subsets of $S \times S$.

Truth is defined in terms of the relation $\models$. We write $(M, s) \models \varphi$ to express that φ is true at world s in the Kripke structure M.

Definition 6. *Truth of a formula is given as follows:*

- $(M, s) \models \mathbf{true}$
- $(M, s) \not\models \mathbf{false}$
- $(M, s) \models p$ *if, and only if,* $\pi(s)(p) = true$*, where* $p \in \mathcal{P}$
- $(M, s) \models \neg\varphi$ *if, and only if,* $(M, s) \not\models \varphi$
- $(M, s) \models (\varphi \wedge \psi)$ *if, and only if,* $(M, s) \models \varphi$ *and* $(M, s) \models \psi$
- $(M, s) \models (\varphi \vee \psi)$ *if, and only if,* $(M, s) \models \varphi$ *or* $(M, s) \models \psi$
- $(M, s) \models (\varphi \Rightarrow \psi)$ *if, and only if,* $(M, s) \models \neg\varphi$ *or* $(M, s) \models \psi$
- $(M, s) \models \boxed{i}\varphi$ *if, and only if, for all* t*, such that* $(s, t) \in \mathcal{R}_i$*,* $(M, t) \models \varphi$*.*

Formulae are interpreted with respect to the distinguished world s_0. Intuitively, s_0 is the world from which we start reasoning. Let $M = \langle \mathcal{S}, \pi, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ be a Kripke structure. Thus, a formula φ is said to be *satisfiable in* M if $(M, s_0) \models \varphi$; it is said to be *satisfiable* if there is a model M such that $(M, s_0) \models \varphi$; and it is said to be *valid* if for all models M then $(M, s_0) \models \varphi$.

3 A Normal Form for $\mathsf{K}_{(n)}$

Formulae in the language of $\mathsf{K}_{(n)}$ can be transformed into a normal form called Separated Normal Form for Normal Logics (SNF_K). We introduce a nullary connective **start**, in order to represent the world from which we start reasoning. Formally, we have that $(M, s) \models \mathbf{start}$ if, and only if, $s = s_0$. A formula in SNF_K

is represented by a conjunction of clauses, which are true at all states, that is, they have the general form

$$\Box^* \bigwedge_i A_i$$

where A_i is a clause and $\Box^*$, the universal operator is defined as:

$(M, s) \models \Box^*\varphi$ if, and only if, $(M, s) \models \varphi$ and for all s' such that $(s, s') \in \mathcal{R}_i$, for some $i \in \mathcal{A}, (M, s') \models \Box^*\varphi$.

Observe that φ holds at the actual world s and at every world reachable from s, where reachability is defined in the usual way. That is, let M be a model and u and u' be worlds in M. Then u' is reachable from u if, and only if, either (i) $(u, u') \in \mathcal{R}_i$ for some agent $i \in \mathcal{A}$; or (ii) there is a world u'' in M such that u'' is reachable from u and u' is reachable from u''. The universal operator, which surrounds all clauses, ensures that the *translation* of a formula is true at all worlds. Clauses are in one of the following forms:

- Initial clause $\quad \mathbf{start} \Rightarrow \bigvee_{b=1}^{r} l_b$
- Literal clause $\quad \mathbf{true} \Rightarrow \bigvee_{b=1}^{r} l_b$
- $\boxed{i}$-clause $\quad l \Rightarrow m_i$

where l and any l_b are literals and m_i is a modal literal containing a $\boxed{i}$ or a $\neg\boxed{i}$ operator. In general, that is, when we do not need to specify a particular agent, we often write *modal clause* to refer to a $\boxed{i}$-clause.

3.1 Transformation into Normal Form

The translation to SNF_K uses the renaming technique [8], where complex subformulae are replaced by new propositional symbols and the truth of these new symbols is linked to the formulae that they replaced in all states. The translation into SNF_K of a given formula φ of $\mathsf{K}_{(n)}$ is given by the following transformation functions, τ_0 and τ_1, where x is a new propositional symbol:

$$\tau_0(\varphi) = \Box^*(\mathbf{start} \Rightarrow x) \wedge \tau_1(\Box^*(x \Rightarrow \varphi))$$

The function τ_0 is used to anchor the meaning of φ to the initial world, where the formula is evaluated. The function τ_1 proceeds with the translation, removing classical operators, by means of classical rewriting operations, and replacing complex formulae which appear in the scope of the $\boxed{i}$ operator, by means of renaming. The next rewriting rules deal with classical operators (where A and B are formulae, and x is the propositional symbol introduced by the function τ_0):

$$\begin{aligned}
\tau_1(\Box^*(x \Rightarrow \neg\neg A)) &= \tau_1(\Box^*(x \Rightarrow A)) \\
\tau_1(\Box^*(x \Rightarrow (A \land B))) &= \tau_1(\Box^*(x \Rightarrow A)) \land \tau_1(\Box^*(x \Rightarrow B)) \\
\tau_1(\Box^*(x \Rightarrow (A \Rightarrow B))) &= \tau_1(\Box^*(x \Rightarrow \neg A \lor B)) \\
\tau_1(\Box^*(x \Rightarrow \neg(A \land B))) &= \tau_1(\Box^*(x \Rightarrow \neg A \lor \neg B)) \\
\tau_1(\Box^*(x \Rightarrow \neg(A \Rightarrow B))) &= \tau_1(\Box^*(x \Rightarrow A)) \land \tau_1(\Box^*(x \Rightarrow \neg B)) \\
\tau_1(\Box^*(x \Rightarrow \neg(A \lor B))) &= \tau_1(\Box^*(x \Rightarrow \neg A)) \land \tau_1(\Box^*(x \Rightarrow \neg B))
\end{aligned}$$

We rename complex subformulae enclosed in a modal operator as follows, where y is a new proposition and A is not a literal.

$$\begin{aligned}
\tau_1(\Box^*(x \Rightarrow \boxed{i} A)) &= \tau_1(\Box^*(x \Rightarrow \boxed{i} y)) \land \tau_1(\Box^*(y \Rightarrow A)) \\
\tau_1(\Box^*(x \Rightarrow \neg\boxed{i} A)) &= \tau_1(\Box^*(x \Rightarrow \neg\boxed{i}\neg y)) \land \tau_1(\Box^*(y \Rightarrow \neg A))
\end{aligned}$$

Next we use renaming on formulae whose right-hand side has disjunction as its main operator but may not be in the correct form (where y is a new proposition, D is a disjunction of formulae, A is not a literal or an implication, and D' and D'' are formulae):

$$\begin{aligned}
\tau_1(\Box^*(x \Rightarrow D \lor (D' \Rightarrow D''))) &= \tau_1(\Box^*(x \Rightarrow D \lor \neg D' \lor D'')) \\
\tau_1(\Box^*(x \Rightarrow D \lor A)) &= \tau_1(\Box^*(x \Rightarrow D \lor y)) \land \tau_1(\Box^*(y \Rightarrow A))
\end{aligned}$$

Finally, we rewrite formulae whose right-hand side is a disjunction of literals into clause form, that is, as an implication. Modal clauses whose right-hand side is a modal literal are already in the normal form, so no further transformation is required. Note that each modal clause contains only one modal literal. So, the different contexts belonging to different agents are already separated at the end of the translation and we do not require further renaming as in [1].

$$\tau_1(\Box^*(x \Rightarrow D)) = \begin{cases} \Box^*(\mathbf{true} \Rightarrow \neg x \lor D) & \text{if } D \text{ is a disjunction of literals} \\ \Box^*(x \Rightarrow D) & \text{if } D \text{ is a modal literal} \end{cases}$$

As an example, the translation of $\boxed{1}(a \Rightarrow b) \Rightarrow (\boxed{1}a \Rightarrow \boxed{1}b)$ is given by:

$$\begin{aligned}
\tau_0(\boxed{1}(a \Rightarrow b) \Rightarrow (\boxed{1}a \Rightarrow \boxed{1}b)) = {} & \Box^*(\mathbf{start} \Rightarrow t_1) \land \\
& \tau_1(\Box^*(t_1 \Rightarrow \boxed{1}(a \Rightarrow b) \Rightarrow (\boxed{1}a \Rightarrow \boxed{1}b)))
\end{aligned}$$

where

$$\tau_1(\Box^*(t_1 \Rightarrow \boxed{1}(a \Rightarrow b) \Rightarrow (\boxed{1}a \Rightarrow \boxed{1}b))) =$$

$= \tau_1(\Box^*(t_1 \Rightarrow \neg(\boxed{1}(a \Rightarrow b)) \vee (\boxed{1}a \Rightarrow \boxed{1}b)))$

$= \tau_1(\Box^*(t_1 \Rightarrow t_2 \vee \neg\boxed{1}a \vee \boxed{1}b)) \wedge \tau_1(\Box^*(t_2 \Rightarrow \neg(\boxed{1}(a \Rightarrow b))))$

$= \tau_1(\Box^*(t_1 \Rightarrow t_2 \vee t_3 \vee t_4)) \wedge \tau_1(\Box^*(t_2 \Rightarrow \neg\boxed{1}\neg t_5)) \wedge \tau_1(\Box^*(t_3 \Rightarrow \neg\boxed{1}a)) \wedge \tau_1(\Box^*(t_4 \Rightarrow \boxed{1}b)) \wedge \tau_1(\Box^*(t_5 \Rightarrow \neg(a \Rightarrow b)))$

$= \Box^*(\mathbf{true} \Rightarrow \neg t_1 \vee t_2 \vee t_3 \vee t_4) \wedge \Box^*(t_2 \Rightarrow \neg\boxed{1}\neg t_5) \wedge \Box^*(t_3 \Rightarrow \neg\boxed{1}a) \wedge \Box^*(t_4 \Rightarrow \boxed{1}b) \wedge \tau_1(\Box^*(t_5 \Rightarrow a)) \wedge \tau_1(\Box^*(t_5 \Rightarrow \neg b)))$

$= \Box^*(\mathbf{true} \Rightarrow \neg t_1 \vee t_2 \vee t_3 \vee t_4) \wedge \Box^*(t_2 \Rightarrow \neg\boxed{1}\neg t_5) \wedge \Box^*(t_3 \Rightarrow \neg\boxed{1}a) \wedge \Box^*(t_4 \Rightarrow \boxed{1}b) \wedge \Box^*(\mathbf{true} \Rightarrow \neg t_5 \vee a) \wedge \Box^*(\mathbf{true} \Rightarrow \neg t_5 \vee \neg b)$

The translation procedure results in 7 clauses: one initial, three literal, and three modal clauses. Note also that the new propositional symbols t_i, $(1 \leq i \leq 5)$, were introduced during renaming of complex formula: either a disjunct which is not a literal or a complex formula inside the scope of a modal operator.

3.2 Correctness of Translation

In order to prove that the translation into normal form is satisfiability preserving, we need to prove the following lemmas. Firstly, we show that if the translated formula is satisfiable in a model, then the original formula is also satisfiable.

Lemma 1. *Let φ be a formula in $\mathsf{K}_{(n)}$, and M be a model. If $M \models \tau_1(\Box^*(x \Rightarrow \varphi))$, then $M \models \Box^*(x \Rightarrow \varphi)$, where x is a propositional symbol.*

Proof. By induction on the structure of φ. In the following s is a world in M. For the base cases, φ is a disjunction of literals or a modal literal. The latter is obvious. We show that if D is a disjunction of literals and $(M, s) \models \tau_1(\Box^*(x \Rightarrow D))$, then $(M, s) \models \Box^*(x \Rightarrow D)$. If $(M, s) \models \tau_1(\Box^*(x \Rightarrow D))$, then, by the transformation givent by τ_1, $(M, s) \models \Box^*(\mathbf{true} \Rightarrow \neg x \vee D)$. By semantics of the universal operator, for any world t reachable from s, we have $(M, t) \models (\mathbf{true} \Rightarrow \neg x \vee D)$. By propositional reasoning $(M, t) \models (\neg\mathbf{true} \vee \neg x \vee D)$, which simplifies to $(M, t) \models (\neg x \vee D)$, as $\neg\mathbf{true}$ cannot be satisfied. Again, by propositional reasoning, $(M, t) \models (x \Rightarrow D)$. By semantics of $\Box^*$, as t is any world reachable from s, we have that $(M, s) \models \Box^*(x \Rightarrow D)$.

We only prove the case where φ is of the form $\boxed{i}A$, where A is not a literal, that is, if $(M, s) \models \tau_1(\Box^*(x \Rightarrow \boxed{i}A))$, we show that $(M, s) \models \Box^*(x \Rightarrow \boxed{i}A)$. The cases for other operators are similar. If $(M, s) \models \tau_1(\Box^*(x \Rightarrow \boxed{i}A))$, we first show that $(M, s) \models \tau_1(\Box^*(x \Rightarrow \boxed{i}y)) \wedge \tau_1(\Box^*(y \Rightarrow A))$. If $(M, s) \models \tau_1(\Box^*(x \Rightarrow \boxed{i}y))$, then (i) $(M, s) \models \Box^*(x \Rightarrow \boxed{i}y)$ (as this is a base case). If $(M, s) \models \tau_1(\Box^*(y \Rightarrow A))$, by induction hypothesis, (ii) $(M, s) \models \Box^*(y \Rightarrow A)$. Let t be a world in M, such that $t = s$ or t is reachable from s. From (i), (ii), the semantics of $\Box^*$, and propositional reasoning, we have $(M, t) \models ((x \Rightarrow \boxed{i}y) \wedge (y \Rightarrow A))$. Also, if $(t, t') \in \mathcal{R}_i$, for some t' in M, by semantics of $\boxed{i}$, we have that if $(M, t) \models x$, then $(M, t') \models y$. As t' is also reachable from s, from (ii), by propositional reasoning, we have $(M, t') \models A$. Thus, by semantics of necessitation, $(M, t) \models (x \Rightarrow \boxed{i}A)$, for all t reachable from s. By semantics of $\Box^*$, we have $(M, s) \models \Box^*(x \Rightarrow \boxed{i}A)$. $\square$

Lemma 2. *Let φ be a formula in $\mathsf{K}_{(n)}$, M be a model. If $M \models \tau_0(\varphi)$, then $M \models \varphi$.*

Proof. Let s be a world in M. If $(M, s) \models \tau_0(\varphi)$, we first show that $(M, s) \models \Box^*(\mathbf{start} \Rightarrow x) \land \tau_1(\Box^*(x \Rightarrow \varphi))$. By semantics of $\Box^*$, $(M, s) \models (\mathbf{start} \Rightarrow x)$. Suppose that $(M, s) \models \mathbf{start}$ (otherwise we are done). Then by semantics of implication, (i) $(M, s) \models x$. By Lemma 1, if $(M, s) \models \tau_1(\Box^*(x \Rightarrow \varphi))$, then $(M, s) \models \Box^*(x \Rightarrow \varphi)$. By semantics of $\Box^*$, we have (ii) $(M, s) \models (x \Rightarrow \varphi)$. From (i) and (ii), by propositional reasoning, $(M, s) \models \varphi$. $\square$

Next we show that if a formula is satisfiable, then its translation into normal form also is satisfiable. The next lemma shows that the transformation function τ_1 preserves satisfiability.

Lemma 3. *Let $\Box^*(x \Rightarrow \varphi)$, where x is a propositional symbol. Let $M = \langle \mathcal{S}, \mathcal{R}_1, \ldots, \mathcal{R}_n, \pi \rangle$ be a model. If $M \models \Box^*(x \Rightarrow \varphi)$, then there is a model M' such that $M' \models \tau_1(\Box^*(x \Rightarrow \varphi))$.*

Proof. By induction on the structure of φ. Suppose that $M \models \Box^*(x \Rightarrow \varphi)$, we show how to construct a model M' such that $M' \models \tau_1(\Box^*(x \Rightarrow \varphi))$. The application of τ_1 consists of two main operations: rewriting and renaming. We examine both operations. For the base cases, that is, if φ is a disjunction of literals or a modal literal, we do not introduce new propositional symbols and we only need propositional reasoning to show that taking $M' = M$, we have $M' \models \tau_1(\Box^*(x \Rightarrow \varphi))$.

Assume that φ is of the form $A \land B$. Then $\tau_1(\Box^*(x \Rightarrow (A \land B)))$ is given by $\tau_1(\Box^*(x \Rightarrow A)) \land \tau_1(\Box^*(x \Rightarrow B))$. Clearly, if $M \models (\Box^*(x \Rightarrow (A \land B)))$, then $M \models (\Box^*(x \Rightarrow A))$ and $M \models (\Box^*(x \Rightarrow B))$. By induction hypothesis, taking M' exactly as M, we obtain a model which satisfies the translation, that is, $M' \models \tau_1(\Box^*(x \Rightarrow (A \land B)))$. Proofs for rewriting of other classical operators are similar.

Now assume that φ is of the form $\boxed{i}A$, where A is not a literal. Then the transformation, $\tau_1(\Box^*(x \Rightarrow \boxed{i}A))$, is given by $\Box^*(x \Rightarrow \boxed{i}y) \land \tau_1(\Box^*(y \Rightarrow A))$. Let M' be exactly as M, but $\pi(s')(y) = \mathit{true}$ for all $s' \in \mathcal{S}$ such that $(s, s') \in \mathcal{R}_i$ and $\pi(s)(x) = \mathit{true}$. Let s be any world in M. Thus, if $(M, s) \models x$, because $(M, s') \models y$, for all $(s, s') \in \mathcal{R}_i$, then $(M', s) \models (x \Rightarrow \boxed{i}y)$; if $(M, s) \not\models x$, then $(M', s) \not\models x$ and the implication holds at (M', s). By semantics of $\Box^*$, $M' \models (\Box^*(x \Rightarrow \boxed{i}y))$. For all $s \in \mathcal{S}$, if $(M', s) \models y$ then $(M', s) \models A$, because y and A are true at the worlds which are are accessible from where x holds. Therefore $(M', s) \models (y \Rightarrow A)$. If $(M', s) \not\models y$, then the implication is vacuously true. Thus, by semantics of $\Box^*$, $M' \models \Box^*(y \Rightarrow A)$. By induction hypothesis, M' satisfies the translation of $\Box^*(x \Rightarrow \boxed{i}A)$, that is, $M' \models \Box^*(x \Rightarrow \boxed{i}y) \land \tau_1(\Box^*(y \Rightarrow A))$. The proofs for other cases of renaming are similar. $\square$

Lemma 4. *Let φ be a formula in $\mathsf{K}_{(n)}$. Let $M = \langle \mathcal{S}, \mathcal{R}_1, \ldots, \mathcal{R}_1, \pi \rangle$ be a model. If $M \models \varphi$, then there is a model M' such that $M' \models \tau_0(\varphi)$.*

Proof. Let $s_0 \in \mathcal{S}$, be the initial world in M. Suppose $M \models \varphi$. Construct a model M' that is identical to M, but where $M' \models \textbf{start}$ if, and only if, $M' \models x$, that is $\pi(s)(x) = \mathit{true}$ if, and only if, $s = s_0$, for $s \in \mathcal{S}$. Clearly, $M' \models \Box^*(\textbf{start} \Rightarrow x) \wedge \Box^*(x \Rightarrow \varphi)$. Given this and, by Lemma 3, it is possible to construct a new model for every application of τ_1, then there is a model such that $M' \models \Box^*(\textbf{start} \Rightarrow x) \wedge \tau_1(\Box^*(x \Rightarrow \varphi))$, that is, $M' \models \tau_0(\varphi)$. $\square$

Thus, a formula is satisfiable if, and only if, its translation into the normal form is satisfiable.

Theorem 1. *Let φ be a formula in $\mathsf{K}_{(n)}$ and M a model. $M \models \varphi$ if, and only if, there is a models M' such that $M' \models \tau_0(\varphi)$.*

Proof. By Lemmas 2 and 4. $\square$

4 Anti-Prenexing

Anti-prenexing has been used in first-order theorem proving as a step applied before skolemization, in order to achieve a better set of clauses [2]. Similarly to first-order, anti-prenexing in the modal context means that all modal operators are moved inwards the formula as far as possible, whilst preserving satisfiability. In the weakest normal logic, $\mathsf{K}_{(n)}$, that means that we can distribute the necessity operator, $\Box$, over conjunctive formulae; and the possibility operator, $\neg\Box\neg$, over disjunctive formulae. The definition of the anti-prenex normal form is given below.

Definition 7. *A **modal term** is a literal or formula of the form $M_1 \dots M_k l$, where l is a literal and M_i, $1 \leq i \leq k$, is $\boxed{j}$ or $\neg\boxed{j}$ for some $j \in \mathcal{A}$.*

Note that aliteral l, which is not preceded by any modal operator, is also a modal term.

Definition 8. *Let φ and ψ be formula in $\mathsf{WFF}_{\mathsf{K}_{(n)}}$. A formula χ is in Anti-Prenex Normal Form (APNF) if, and only if,*

1. *χ is a modal term; or*
2. *χ is of the form $\neg\varphi$, $(\varphi\wedge\psi)$, $(\varphi\vee\psi)$ or $(\varphi \Rightarrow \psi)$, and φ and ψ are in APNF;*
3. *χ is of the form $\boxed{i}\varphi$, φ is disjunctive, and/or φ is in APNF; or*
4. *χ is of the form $\neg\boxed{i}\varphi$, φ is conjunctive, and/or φ is in APNF.*

The following lemma shows that any formula can be transformed into APNF.

Lemma 5. *Let φ be a formula in $\mathsf{K}_{(n)}$ and M a model in $\mathsf{K}_{(n)}$. Then there is a formula φ' in APNF such that if $M \models \varphi$ then $M \models \varphi'$.*

Proof. The following schemata are theorems of $\mathsf{K}_{(n)}$:

1. $\boxed{i}(\varphi \wedge \psi) \Leftrightarrow (\boxed{i}\varphi \wedge \boxed{i}\psi)$

2. $\boxed{i}\neg(\varphi \Rightarrow \psi) \Leftrightarrow (\boxed{i}\varphi \wedge \boxed{i}\neg\psi)$
3. $\boxed{i}\neg(\varphi \vee \psi) \Leftrightarrow (\boxed{i}\neg\varphi \wedge \boxed{i}\neg\psi)$
4. $\neg\boxed{i}\neg(\varphi \Rightarrow \psi) \Leftrightarrow (\boxed{i}\varphi \Rightarrow \neg\boxed{i}\neg\psi)$
5. $\neg\boxed{i}\neg(\varphi \vee \psi) \Leftrightarrow (\neg\boxed{i}\neg\varphi \vee \neg\boxed{i}\neg\psi)$
6. $\neg\boxed{i}(\varphi \wedge \psi) \Leftrightarrow (\neg\boxed{i}\varphi \vee \neg\boxed{i}\psi)$

$\square$

The transformation into SNF_K consists of two steps: transforming the formulae into anti-prenex, as defined below, and then applying the transformation function given in Subsection 3.1. Firstly, based on the schemata presented in Lemma 5, we define a function $\alpha(\varphi)$, where φ is a formula, which produces the anti-prenex normal form of φ. The base case occurs when the formula A is already in APNF, that is, A is a modal term. In this case, $\alpha(A) = A$. If the main operator is modal, we only apply the transformation function to formula which satisfies the equivalences in Lemma 5, that is, in the following cases:

$$\begin{aligned}
\alpha(\boxed{i}(A \wedge B)) &= \alpha(\boxed{i}A \wedge \boxed{i}B)\\
\alpha(\boxed{i}\neg(A \Rightarrow B)) &= \alpha(\boxed{i}A \wedge \boxed{i}\neg B)\\
\alpha(\boxed{i}\neg(A \vee B)) &= \alpha(\boxed{i}\neg A \wedge \boxed{i}\neg B)\\
\alpha(\neg\boxed{i}\neg(A \Rightarrow B)) &= \alpha(\boxed{i}A \Rightarrow \neg\boxed{i}\neg B)\\
\alpha(\neg\boxed{i}\neg(A \vee B)) &= \alpha(\neg\boxed{i}\neg A \vee \neg\boxed{i}\neg B)\\
\alpha(\neg\boxed{i}(A \wedge B)) &= \alpha(\neg\boxed{i}A \vee \neg\boxed{i}B)
\end{aligned}$$

If we have two consecutive modal operators, the function is applied recursively, where A is of the form $\boxed{j}B$ or $\neg\boxed{j}B$, for any $j \in \mathcal{A}$:

$$\begin{aligned}
\alpha(\boxed{i}A) &= \alpha(\boxed{i}\alpha(A))\\
\alpha(\neg\boxed{i}A) &= \alpha(\neg\boxed{i}\alpha(A))
\end{aligned}$$

If the main operator is a modal operator, but the formula inside its scope is not one of the above, we apply the anti-prenexing function to this formula, that is:

$$\begin{aligned}
\alpha(\boxed{i}A) &= \boxed{i}\alpha(A)\\
\alpha(\neg\boxed{i}A) &= \neg\boxed{i}\alpha(A)
\end{aligned}$$

When the main operator is classical, the transformation function is also applied recursively. Note that when the polarity of a subformula is negative, we rewrite the formula in order to make this explicit.

$$\begin{aligned}
\alpha(\neg\neg A) &= \alpha(A) & \alpha(\neg(A \Rightarrow B)) &= (\alpha(A) \wedge \alpha(\neg B))\\
\alpha(A \Rightarrow B) &= \alpha(\neg A) \vee \alpha(B) & \alpha(\neg(A \wedge B)) &= (\alpha(\neg A) \vee \alpha(\neg B))\\
\alpha(A \wedge B) &= \alpha(A) \wedge \alpha(B) & \alpha(\neg(A \vee B)) &= (\alpha(\neg A) \wedge \alpha(\neg B))\\
\alpha(A \vee B) &= \alpha(A) \vee \alpha(B) & &
\end{aligned}$$

The proof that this transformation is correct and satisfiability can be obtained as in Subsection 3.2 for translation into SNF_K and will not be presented here.

As an example, the APNF of $\boxed{i}(a \wedge \boxed{i}(b \wedge \boxed{i}c))$ is given by:

$$\begin{aligned}
&\alpha(\boxed{i}(a \wedge \boxed{i}(b \wedge \boxed{i}c))) = \\
&\quad = \alpha(\boxed{i}a \wedge \boxed{i}\,\boxed{i}(b \wedge \boxed{i}c)) \\
&\quad = \alpha(\boxed{i}a) \wedge \alpha(\boxed{i}\,\boxed{i}(b \wedge \boxed{i}c)) \\
&\quad = \boxed{i}a \wedge \alpha(\boxed{i}\alpha(\boxed{i}(b \wedge \boxed{i}c))) \\
&\quad = \boxed{i}a \wedge \alpha(\boxed{i}\alpha(\boxed{i}b \wedge \boxed{i}\,\boxed{i}c)) \\
&\quad = \boxed{i}a \wedge \alpha(\boxed{i}(\alpha(\boxed{i}b) \wedge \alpha(\boxed{i}\,\boxed{i}c))) \\
&\quad = \boxed{i}a \wedge \alpha(\boxed{i}(\boxed{i}b \wedge \boxed{i}\,\boxed{i}c)) \\
&\quad = \boxed{i}a \wedge \alpha(\boxed{i}\,\boxed{i}b \wedge \boxed{i}\,\boxed{i}\,\boxed{i}c) \\
&\quad = \boxed{i}a \wedge \alpha(\boxed{i}\,\boxed{i}b) \wedge \alpha(\boxed{i}\,\boxed{i}\,\boxed{i}c) \\
&\quad = \boxed{i}a \wedge \boxed{i}\,\boxed{i}b \wedge \boxed{i}\,\boxed{i}\,\boxed{i}c
\end{aligned}$$

whose transformation into the normal form is:

$$\Box^*(\mathbf{start} \Rightarrow x) \wedge \Box^*(x \Rightarrow \boxed{i}a) \wedge \Box^*(x \Rightarrow \boxed{i}y) \wedge \Box^*(y \Rightarrow \boxed{i}b) \wedge \\ \Box^*(x \Rightarrow \boxed{i}z) \wedge \Box^*(z \Rightarrow \boxed{i}w) \wedge \Box^*(w \Rightarrow \boxed{i}c).$$

5 Simplification Rules

The anti-prenexing pre-processing of a formula may result in fewer or shorter clauses. For instance, consider the formula $\boxed{i}(a \wedge b)$. Transformation into SNF_K results in four clauses ($\Box^*(\mathbf{start} \Rightarrow x)$, $\Box^*(x \Rightarrow \boxed{i}y)$, $\Box^*(\mathbf{true} \Rightarrow \neg y \vee a)$, and $\Box^*(\mathbf{true} \Rightarrow \neg y \vee b)$), whilst transformation into the normal form preceded by anti-prenexing results in three clauses ($\Box^*(\mathbf{start} \Rightarrow x)$, $\Box^*(x \Rightarrow \boxed{i}a)$, and $\Box^*(x \Rightarrow \boxed{i}b)$). However, this is not always the case. Depending on the nesting of modal operators in the original formula, the number of clauses generated after anti-prenexing and translation into SNF_K can be significantly larger than by applying the transformation into SNF_K alone. The reason is that the modal operator that had appeared only once in the formula now has several copies distributed over subformulae.

However, when applied together with simplification, anti-prenexing may reduce the number of clauses, by collapsing of nested modal operators in the original formula. Obviously, this depends on the particular normal modal logic we are considering. We discuss in this section the simplification rules that could be applied together with anti-prenexing, before transformation into SNF_K, in the case of $\mathsf{KTD45}_{(n)}$ and $\mathsf{KD45}_{(n)}$. The first normal modal logic, also known as $\mathsf{S5}_{(n)}$ – the logic of knowledge for multiple agents – is axiomatisable by the schemata **K**, **T**, **D**, **4**, and **5** and the rules of inference: *modus ponens* (from $\vdash \varphi$ and $\vdash (\varphi \Rightarrow \psi)$ infer $\vdash \psi$) and *modal necessitation rule* (from $\vdash \varphi$ infer $\vdash \boxed{i}\varphi$). The logics $\mathsf{KD45}_{(n)}$, known as the logic of belief for multiple agents, is axiomatisable by the schemata **K**, **D**, **4**, and **5** and the rules of inference *modus ponens* and *modal necessitation rule*. As the schemata $\boxed{i}\,\boxed{i}\varphi \Leftrightarrow \boxed{i}\varphi$, $\boxed{i}\neg\boxed{i}\varphi \Leftrightarrow \neg\boxed{i}\varphi$, $\neg\boxed{i}\neg\boxed{i}\varphi \Leftrightarrow \boxed{i}\varphi$, $\neg\boxed{i}\,\boxed{i}\varphi \Leftrightarrow \neg\boxed{i}\varphi$ are valid in $\mathsf{KTD45}_{(n)}$ and in $\mathsf{KD45}_{(n)}$, we extend the anti-prenexing function in the obvious way:

$$\begin{aligned}
\alpha(\boxed{i}\,\boxed{i}\varphi) &= \alpha(\boxed{i}\varphi)\\
\alpha(\boxed{i}\neg\boxed{i}\varphi) &= \alpha(\neg\boxed{i}\varphi)\\
\alpha(\neg\boxed{i}\neg\boxed{i}\varphi) &= \alpha(\boxed{i}\varphi)\\
\alpha(\neg\boxed{i}\,\boxed{i}\varphi) &= \alpha(\neg\boxed{i}\varphi)
\end{aligned}$$

We note that we only apply this simplification when the modal operators have the same index. Other simplification rules could also be introduced, but we chose not to do this and, instead, preserving some of the structure of the formula. Using these simplification rules, the anti-prenex normal form of the previous example, i.e. $\boxed{i}(a \wedge \boxed{i}(b \wedge \boxed{i}c))$, is $\boxed{i}a \wedge \boxed{i}b \wedge \boxed{i}c$, which has the same size as the original formula. The translation into SNF_K results, however, in four clauses. Table 1 show the three transformations for comparison.

$\boxed{i}(a \wedge \boxed{i}(b \wedge \boxed{i}c))$

SNF_K	AP + SNF_K	AP + SIMP + SNF_K
$\boxed{i}(a \wedge \boxed{i}(b \wedge \boxed{i}c))$	$\boxed{i}a \wedge \boxed{i}\,\boxed{i}b \wedge \boxed{i}\,\boxed{i}\,\boxed{i}c$	$\boxed{i}a \wedge \boxed{i}b \wedge \boxed{i}c$
1. $\mathbf{start} \Rightarrow x$	1. $\mathbf{start} \Rightarrow x$	1. $\mathbf{start} \Rightarrow x$
2. $x \Rightarrow \boxed{i}y$	2. $x \Rightarrow \boxed{i}a$	2. $x \Rightarrow \boxed{i}a$
3. $\mathbf{true} \Rightarrow \neg y \vee a$	3. $x \Rightarrow \boxed{i}y$	3. $x \Rightarrow \boxed{i}b$
4. $y \Rightarrow \boxed{i}z$	4. $y \Rightarrow \boxed{i}b$	4. $y \Rightarrow \boxed{i}c$
5. $\mathbf{true} \Rightarrow \neg z \vee b$	5. $x \Rightarrow \boxed{i}z$	
6. $z \Rightarrow \boxed{i}c$	6. $z \Rightarrow \boxed{i}w$	
	7. $w \Rightarrow \boxed{i}c$	

Table 1. Translation (from left to right) without Anti-Prenexing, after Anti-Prenexing, and after Anti-Prenexing and Simplification.

Note that no simplification rule could be applied to the original formula. By moving the modal operators inwards the formula, simplification could be applied, resulting in fewer and shorter clauses.

6 Prenexing

Similarly to first-order logic, prenexing in the modal context means to pull modal operators as far as possible outwards the formula. It is well-known that formulae in $\mathsf{KTD45}_{(1)}$ can be transformed into a formula without nesting of modal operators (see [5], for instance). As we are interested in a more general form of prenexing than that given for $\mathsf{KTD45}_{(1)}$, we say that a formula is in prenex normal form if it corresponds to the inverse of the transformation into anti-prenexing. Thus, the transformation is justified by the same equivalences appearing in Lemma 5. Our definition of the prenexing function is similar to that given in Section 4 and will not be presented here. Instead, in this section, we give the motivation for using both techniques together with simplification for $\mathsf{KTD45}_{(n)}$ and $\mathsf{KD45}_{(n)}$.

When a formula φ is transformed into APNF, the nesting of modal operators is made explicit and can be easily simplified. On the other hand, several copies of a modal operator may now appear in the formula. By performing the transformation into prenex normal form, after anti-prenexing and simplification, we try to remove such copies and make the formula shorter. We note that the order in which the transformations are applied is important. Consider, for instance, the formula given in previous examples, that is, $\boxed{i}(a \wedge \boxed{i}(b \wedge \neg\boxed{i}\neg c))$. This formula is in prenex normal form, as we cannot apply any of the equivalences given in Lemma 5 to move the modal operators outwards the formula. However, if we apply anti-prenexing with simplification, we obtain, as seen before, the formula $\boxed{i}a \wedge \boxed{i}b \wedge \boxed{i}c$. The result of applying the prenex function is $\boxed{i}(a \wedge b \wedge c)$, which is shorter than both the original formula and the one obtained after anti-prenexing with simplification. In this case, however, the transformation into SNF_K results in one more clause. We give below an example where the number of clauses is smaller than that produced by the other methods without prenexing:

Formula	$\neg\boxed{i}\neg(a \vee \neg\boxed{i}\neg(b \vee \neg\boxed{i}\neg c))$
	$\downarrow$
Anti-Prenexing	$\neg\boxed{i}\neg a \vee \neg\boxed{i}\neg\neg\boxed{i}\neg b \vee \neg\boxed{i}\neg\neg\boxed{i}\neg\neg\boxed{i}\neg c$
	$\downarrow$
Anti-Prenexing + Simplification	$\neg\boxed{i}\neg a \vee \neg\boxed{i}\neg b \vee \neg\boxed{i}\neg c$
	$\downarrow$
Prenex	$\neg\boxed{i}\neg(a \vee b \vee c)$
	$\downarrow$
	1. $\Box^*(\mathbf{start} \Rightarrow x)$
SNF_K	2. $\Box^*(\mathbf{true} \Rightarrow \neg\boxed{i}\neg y)$
	3. $\Box^*(\mathbf{true} \Rightarrow \neg y \vee a \vee b \vee c)$

As a final example, consider the formula $\neg\boxed{i}\neg(\boxed{i}a \wedge \boxed{i}b)$, which is already in APNF, as the modal operator $\neg\boxed{i}\neg$ cannot be distributed over conjunctions. Also, no simplification rule can be applied to the formula. After applying prenexing, however, we obtain $\neg\boxed{i}\neg\boxed{i}(a \wedge b)$, which simplifies to $\boxed{i}(a \wedge b)$. We do not apply anti-prenexing again, which would result in a shorter translation into SNF_K, as discussed at the beginning of Section 5. Nevertheless, the resulting formula is half the size of the original one and the set of clauses is also smaller.

7 Experimental Results

The examples given here have only the purpose of illustrating the techniques. We cannot prove, in general, that by applying those techniques we will obtain a better set of clauses. In order to have a measure of how anti-prenexing and prenexing behave in comparison to translation to SNF_K alone, we have performed tests using formulae from [4].

The program takes a modal formula and returns its size and number of literals. It also returns the size and number of literals after transforming the formula into SNF_K alone, into SNF_K preceded by anti-prenexing, and into SNF_K preceded by anti-prenexing with simplification. We are currently working on the implementation of prenexing.

Table 2 shows the output from running the program over formulae from the benchmark `s4_45_p.txt`, which contains problems in $\mathsf{KT4}_{(1)}$ which are provable in both $\mathsf{KT4}_{(1)}$ and $\mathsf{KTD45}_{(1)}$. For each problem, identified in the first column, we present the total size of the formula (*Size*) and the number of different literals (*Lits*). Columns 2 and 3 refer to the original formula. Columns 4 and 5 show the result for transformation into SNF_K alone. Columns 6 to 8 are related to anti-prenexing without simplification, where the first two columns are the size of the problem after transforming into anti-prenexing, and the other two columns are the result of transforming into SNF_K. The last four columns contain the result for anti-prenexing together with simplification: their contents are similar to those for anti-prenexing without simplification. The table shows that transformation into SNF_K preceded by anti-prenexing with simplification perform better than the other two methods. Other experimental results can be found in Appendix A.

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	119	3	207	27	99	3	189	27	77	3	141	19
2	313	3	531	50	257	3	479	50	213	3	383	44
3	581	3	979	87	473	3	877	87	407	3	733	77
4	923	3	1551	122	747	3	1383	150	659	3	1191	130
5	1339	3	2247	183	1079	3	1997	184	969	3	1757	186
6	1829	3	3067	256	1469	3	2719	296	1337	3	2431	264
7	2393	3	4011	387	1917	3	3549	387	1763	3	3213	349
8	3031	3	5079	489	2423	3	4487	489	2247	3	4103	438
9	3743	3	6271	544	2987	3	5533	604	2789	3	5101	555
10	4529	3	7587	664	3609	3	6687	665	3389	3	6207	667
11	5389	3	9027	797	4289	3	7949	869	4047	3	7421	809
12	6323	3	10591	1020	5027	3	9319	1020	4763	3	8743	954
13	7331	3	12279	1182	5823	3	10797	1182	5537	3	10173	1099
14	8413	3	14091	1265	6677	3	12383	1357	6369	3	11711	1280
15	9569	3	16027	1445	7589	3	14077	1446	7259	3	13357	1448
16	10799	3	18087	1638	8559	3	15879	1742	8207	3	15111	1654
17	12103	3	20271	1953	9587	3	17789	1953	9213	3	16973	1859
18	13481	3	22579	2175	10673	3	19807	2175	10277	3	18943	2060
19	14933	3	25011	2286	11817	3	21933	2410	11399	3	21021	2305
20	16459	3	27567	2526	13019	3	24167	2527	12579	3	23207	2529
21	18059	3	30247	2779	14279	3	26509	2915	13817	3	25501	2799

Table 2. Results for Transformations of Formulae in the Logic Workbench

8 Conclusions

In this paper we have presented an algorithm for transforming any normal modal formula into a normal form. This can be done because the transformation is based on valid schemata of the weakest of the normal modal logics, namely $\mathsf{K}_{(n)}$. Also, we investigate how the use of anti-prenexing and prenexing can help in obtaining a better transformation. Combined with simplification rules, these methods seem to produce smaller clause sets for problems from some normal logics.

There is no way of defining which is the best normal form, in general. Here we focused on the size of the transformed problem as a measure for determining whether the transformation is good. However, we also intend to investigate other parameters, as for instance the number of clauses and their sizes, as well as how efficiently a real theorem-prover responds to those different transformations. As anti-prenexing with simplification moves the modal operators inwards the formula, those operators are usually applied to simpler formulae, indicating that we could have less resolution steps applied to a clause set.

Simplification is an important step in the translation algorithm, but, as discussed before, it cannot be applied to all normal modal logics. We have shown the simplification rules for $\mathsf{KTD45}_{(n)}$ and $\mathsf{KD45}_{(n)}$. The equivalences $\boxed{i}\neg\boxed{i}\varphi \Leftrightarrow \neg\boxed{i}\varphi$ and $\neg\boxed{i}\neg\boxed{i}\varphi \Leftrightarrow \boxed{i}\varphi$ are also valid in $\mathsf{K45}_{(n)}$ and $\mathsf{KT4}_{(n)}$ (also known as $\mathsf{S4}_{(n)}$), so the formula resulting from anti-prenexing can be simplified, but not at the same extent as those of $\mathsf{KTD45}_{(n)}$ and $\mathsf{KD45}_{(n)}$. The other modal logics do not admit simplification rules for collapsing of nested operators. In these cases, as our experimental results show, applying anti-prenexing does not seem to be worthwhile. We are currently working on the complexity of the transformation in order to determine precisely when anti-prenexing and prenexing of a formula would result in a better set of clauses. That is, the techniques shown here could be used *selectively* in a similar way as renaming is used [7].

We are currently working on the implementation of the prenexing algorithm. We believe that anti-prenexing together with prenexing and simplification will give us the best result for formulae in $\mathsf{KTD45}_{(n)}$ and $\mathsf{KD45}_{(n)}$. We hope that the same combination will give us better results for the other logics.

Current work also involves the development of the resolution-based methods for each logic. Our intention is that a uniform approach to deal with those logics – from the designing of the normal forms up to the whole proof-method – will facilitate the task of validity checking for combinations of those logics.

Acknowledgements

The first author was supported by CNPq grants CT-INFO 506598/04-7 and Universal 47171/2004-0.

References

1. C. Dixon and M. Fisher. Resolution-Based Proof for Multi-Modal Temporal Logics of Knowledge. In S. Goodwin and A. Trudel, editors, *Proceedings of the Seventh In-*

ternational Workshop on Temporal Representation and reasoning (TIME'00), pages 69–78, Cape Breton, Nova Scotia, Canada, July 2000. IEEE Computer Society Press.
2. U. Egly. On the value of antiprenexing. In F. Pfenning, editor, *Proceedings of the 5th International Conference on Logic Programming and Automated Reasoning*, volume 822 of *LNAI*, pages 69–83, Berlin, July 1994. Springer.
3. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.
4. G. Jaeger, P. Balsiger, A. Heuerding, S. Schwendimann, M. Bianchi, K. Guggisberg, G. Janssen, W. Heinle, F. Achermann, A. D. Boroumand, P. Brambilla, I. Bucher, and H. Zimmermann. LWB–The Logics Workbench 1.1. http://www.lwb.unibe.ch/. University of Berne, Switzerland.
5. J. J. C. Meyer and W. van der Hoek. *Epistemic Logic for Computer Science and Artificial Intelligence*, volume 41 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1995.
6. C. Nalon and C. Dixon. Anti-prenexing and prenexing for modal logics (extended version). Available at http://www.cic.unb.br/∼nalon and http://www.csc.liv.ac.uk/∼clare/cld-pubs/, April 2006.
7. A. Nonnengart and C. Weidenbach. Computing small clause normal forms. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 6, pages 335–367. Elsevier Science B.V., 2001.
8. D. A. Plaisted and S. A. Greenbaum. A Structure-Preserving Clause Form Translation. *Journal of Logic and Computation*, 2:293–304, 1986.
9. A. S. Rao and M. P. Georgeff. Decision procedures for BDI logics. *Journal of Logic and Computation*, 8(3):293–342, 1998.

A Experimental Results

In this appendix we show results from applying the algorithms for anti-prenexing and simplification to some formulae from [4].

A.1 s4-45-n

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	161	2	285	32	131	2	261	32	119	2	225	26
2	412	2	707	47	332	2	639	55	308	2	567	54
3	752	2	1273	96	602	2	1141	101	566	2	1033	98
4	1181	2	1983	148	941	2	1767	148	893	2	1623	158
5	1699	2	2837	204	1349	2	2517	234	1289	2	2337	216
6	2306	2	3835	300	1826	2	3391	320	1754	2	3175	326
7	3002	2	4977	397	2372	2	4389	418	2288	2	4137	406
8	3787	2	6263	530	2987	2	5511	529	2891	2	5223	515
9	4661	2	7693	628	3671	2	6757	628	3563	2	6433	646
10	5624	2	9267	736	4424	2	8127	792	4304	2	7767	756
11	6676	2	10985	911	5246	2	9621	943	5114	2	9225	951
12	7817	2	12847	1072	6137	2	11239	1106	5993	2	10807	1083
13	9047	2	14853	1283	7097	2	12981	1282	6941	2	12513	1257
14	10366	2	17003	1433	8126	2	14847	1433	7958	2	14343	1459
15	11774	2	19297	1593	9224	2	16837	1675	9044	2	16297	1621
16	13271	2	21735	1846	10391	2	18951	1891	10199	2	18375	1901
17	14857	2	24317	2072	11627	2	21189	2119	11423	2	20577	2085
18	16532	2	27043	2361	12932	2	23551	2360	12716	2	22903	2324
19	18296	2	29913	2563	14306	2	26037	2563	14078	2	25353	2597
20	20149	2	32927	2775	15749	2	28647	2883	15509	2	27927	2811
21	22091	2	36085	3106	17261	2	31381	3164	17009	2	30625	3176

A.2 s4-45-p

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	119	3	207	27	99	3	189	27	77	3	141	19
2	313	3	531	50	257	3	479	50	213	3	383	44
3	581	3	979	87	473	3	877	87	407	3	733	77
4	923	3	1551	122	747	3	1383	150	659	3	1191	130
5	1339	3	2247	183	1079	3	1997	184	969	3	1757	186
6	1829	3	3067	256	1469	3	2719	296	1337	3	2431	264
7	2393	3	4011	387	1917	3	3549	387	1763	3	3213	349
8	3031	3	5079	489	2423	3	4487	489	2247	3	4103	438
9	3743	3	6271	544	2987	3	5533	604	2789	3	5101	555
10	4529	3	7587	664	3609	3	6687	665	3389	3	6207	667
11	5389	3	9027	797	4289	3	7949	869	4047	3	7421	809
12	6323	3	10591	1020	5027	3	9319	1020	4763	3	8743	954
13	7331	3	12279	1182	5823	3	10797	1182	5537	3	10173	1099
14	8413	3	14091	1265	6677	3	12383	1357	6369	3	11711	1280
15	9569	3	16027	1445	7589	3	14077	1446	7259	3	13357	1448
16	10799	3	18087	1638	8559	3	15879	1742	8207	3	15111	1654
17	12103	3	20271	1953	9587	3	17789	1953	9213	3	16973	1859
18	13481	3	22579	2175	10673	3	19807	2175	10277	3	18943	2060
19	14933	3	25011	2286	11817	3	21933	2410	11399	3	21021	2305
20	16459	3	27567	2526	13019	3	24167	2527	12579	3	23207	2529
21	18059	3	30247	2779	14279	3	26509	2915	13817	3	25501	2799

A.3 s4-branch-n

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	76	5	240	27	100	5	250	32	100	5	250	32
2	124	7	375	38	164	7	444	44	164	7	444	44
3	172	9	510	48	228	9	587	55	228	9	587	67
4	220	11	645	66	292	11	781	83	292	11	781	89
5	268	13	780	79	356	13	924	106	356	13	924	106
6	316	15	915	92	420	15	1118	117	420	15	1118	124
7	364	17	1050	104	484	17	1261	139	484	17	1261	139
8	412	19	1185	118	548	19	1455	151	548	19	1455	155
9	460	21	1320	129	612	21	1598	168	612	21	1598	172
10	508	23	1455	144	676	23	1792	203	676	23	1792	203
11	556	25	1590	157	740	25	1935	204	740	25	1935	205
12	604	27	1725	170	804	27	2129	222	804	27	2129	222
13	652	29	1860	181	868	29	2272	221	868	29	2272	245
14	700	31	1995	196	932	31	2466	265	932	31	2466	283
15	748	33	2130	209	996	33	2609	296	996	33	2609	296
16	796	35	2265	222	1060	35	2803	295	1060	35	2803	314
17	844	37	2400	235	1124	37	2946	317	1124	37	2946	317
18	892	39	2535	248	1188	39	3140	329	1188	39	3140	333
19	940	41	2670	259	1252	41	3283	346	1252	41	3283	350
20	988	43	2805	274	1316	43	3477	393	1316	43	3477	393
21	1036	45	2940	287	1380	45	3620	382	1380	45	3620	383

A.4 s4-branch-p

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	80	5	253	29	104	5	257	33	104	5	257	33
2	128	7	388	38	168	7	451	49	168	7	451	53
3	176	9	523	55	232	9	594	70	232	9	594	72
4	224	11	658	68	296	11	788	85	296	11	788	89
5	272	13	793	81	360	13	931	99	360	13	931	104
6	320	15	928	92	424	15	1125	120	424	15	1125	124
7	368	17	1063	107	488	17	1268	140	488	17	1268	148
8	416	19	1198	120	552	19	1462	170	552	19	1462	170
9	464	21	1333	133	616	21	1605	176	616	21	1605	186
10	512	23	1468	146	680	23	1799	188	680	23	1799	192
11	560	25	1603	159	744	25	1942	206	744	25	1942	211
12	608	27	1738	170	808	27	2136	227	808	27	2136	231
13	656	29	1873	185	872	29	2279	260	872	29	2279	262
14	704	31	2008	198	936	31	2473	263	936	31	2473	267
15	752	33	2143	211	1000	33	2616	277	1000	33	2616	282
16	800	35	2278	222	1064	35	2810	298	1064	35	2810	302
17	848	37	2413	237	1128	37	2953	318	1128	37	2953	338
18	896	39	2548	250	1192	39	3147	360	1192	39	3147	360
19	944	41	2683	263	1256	41	3290	354	1256	41	3290	376
20	992	43	2818	276	1320	43	3484	370	1320	43	3484	374
21	1040	45	2953	289	1384	45	3627	384	1384	45	3627	389

A.5 s4-grz-n

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	153	4	520	72	186	4	534	74	183	4	504	67
2	175	4	592	78	211	4	605	83	208	4	575	76
3	207	4	684	85	248	4	697	86	245	4	667	84
4	247	4	794	93	294	4	807	99	291	4	777	92
5	295	5	922	102	349	5	935	104	346	5	905	99
6	355	5	1100	112	413	5	1113	116	410	5	1083	115
7	423	5	1296	129	486	5	1309	132	483	5	1279	127
8	499	5	1510	143	568	5	1523	146	565	5	1493	143
9	583	5	1742	156	659	5	1755	162	656	5	1725	157
10	679	5	2024	173	759	5	2037	180	756	5	2007	175
11	783	5	2324	193	868	5	2337	200	865	5	2307	195
12	895	5	2642	213	986	5	2655	220	983	5	2625	215
13	1015	5	2978	233	1113	5	2991	240	1110	5	2961	235
14	1147	5	3364	256	1249	5	3377	264	1246	5	3347	263
15	1287	5	3768	283	1394	5	3781	284	1391	5	3751	280
16	1435	5	4190	304	1548	5	4203	315	1545	5	4173	310
17	1591	5	4630	328	1711	5	4643	340	1708	5	4613	341
18	1759	5	5120	371	1883	5	5133	373	1880	5	5103	368
19	1935	5	5628	384	2064	5	5641	398	2061	5	5611	393
20	2119	5	6154	415	2254	5	6167	425	2251	5	6137	437
21	2311	5	6698	460	2453	5	6711	462	2450	5	6681	458

A.6 s4-grz-p

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	192	5	662	89	231	5	669	89	227	5	639	83
2	213	5	728	93	256	5	734	96	252	5	704	91
3	244	5	814	100	294	5	820	104	290	5	790	98
4	283	5	918	108	342	5	924	109	338	5	894	104
5	330	6	1040	115	400	6	1046	119	396	6	1016	113
6	389	6	1212	123	464	6	1218	129	460	6	1188	124
7	456	6	1402	135	538	6	1408	136	534	6	1378	135
8	531	6	1610	146	622	6	1616	150	618	6	1586	152
9	614	6	1836	170	716	6	1842	171	712	6	1812	165
10	709	6	2112	181	816	6	2118	190	812	6	2088	184
11	812	6	2406	203	926	6	2412	204	922	6	2382	203
12	923	6	2718	215	1046	6	2724	220	1042	6	2694	219
13	1042	6	3048	238	1176	6	3054	239	1172	6	3024	235
14	1173	6	3428	265	1312	6	3434	266	1308	6	3404	261
15	1312	6	3826	279	1458	6	3832	280	1454	6	3802	276
16	1459	6	4242	304	1614	6	4248	305	1610	6	4218	301
17	1614	6	4676	329	1780	6	4682	330	1776	6	4652	326
18	1781	6	5160	358	1952	6	5166	366	1948	6	5136	361
19	1956	6	5662	390	2134	6	5668	409	2130	6	5638	403
20	2139	6	6182	420	2326	6	6188	433	2322	6	6158	434
21	2330	6	6720	463	2528	6	6726	471	2524	6	6696	465

A.7 s4-ipc-n

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	3	2	3	2	4	2	12	4	4	2	12	4
2	16	4	50	9	20	4	51	9	20	4	51	9
3	35	5	97	12	42	5	98	14	42	5	98	15
4	60	6	152	18	70	6	153	20	70	6	153	21
5	91	7	215	24	104	7	216	26	104	7	216	27
6	128	8	286	30	144	8	287	32	144	8	287	33
7	171	9	365	36	190	9	366	38	190	9	366	39
8	220	10	452	42	242	10	453	44	242	10	453	45
9	275	11	547	48	300	11	548	50	300	11	548	51
10	336	12	650	54	364	12	651	56	364	12	651	57
11	403	13	761	60	434	13	762	62	434	13	762	63
12	476	14	880	66	510	14	881	68	510	14	881	69
13	555	15	1007	72	592	15	1008	74	592	15	1008	75
14	640	16	1142	78	680	16	1143	80	680	16	1143	81
15	731	17	1285	84	774	17	1286	86	774	17	1286	87
16	828	18	1436	90	874	18	1437	92	874	18	1437	93
17	931	19	1595	96	980	19	1596	98	980	19	1596	99
18	1040	20	1762	102	1092	20	1763	104	1092	20	1763	105
19	1155	21	1937	108	1210	21	1938	110	1210	21	1938	111
20	1276	22	2120	114	1334	22	2121	116	1334	22	2121	117
21	1403	23	2311	120	1464	23	2312	122	1464	23	2312	123

A.8 s4-ipc-p

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	11	2	44	10	14	2	44	10	14	2	44	10
2	27	3	87	13	33	3	87	13	33	3	87	14
3	49	4	138	17	58	4	138	19	58	4	138	20
4	77	5	197	23	89	5	197	25	89	5	197	26
5	111	6	264	29	126	6	264	31	126	6	264	32
6	151	7	339	35	169	7	339	37	169	7	339	38
7	197	8	422	41	218	8	422	43	218	8	422	44
8	249	9	513	47	273	9	513	49	273	9	513	50
9	307	10	612	53	334	10	612	55	334	10	612	56
10	371	11	719	59	401	11	719	61	401	11	719	62
11	441	12	834	65	474	12	834	67	474	12	834	68
12	517	13	957	71	553	13	957	73	553	13	957	74
13	599	14	1088	77	638	14	1088	79	638	14	1088	80
14	687	15	1227	83	729	15	1227	85	729	15	1227	86
15	781	16	1374	89	826	16	1374	91	826	16	1374	92
16	881	17	1529	95	929	17	1529	97	929	17	1529	98
17	987	18	1692	101	1038	18	1692	103	1038	18	1692	104
18	1099	19	1863	107	1153	19	1863	109	1153	19	1863	110
19	1217	20	2042	113	1274	20	2042	115	1274	20	2042	116
20	1341	21	2229	119	1401	21	2229	121	1401	21	2229	122
21	1471	22	2424	125	1534	22	2424	127	1534	22	2424	128

A.9 s4-md-n

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	3	2	11	5	3	2	11	5	3	2	11	5
2	15	2	42	9	15	2	42	9	12	2	36	8
3	50	2	129	21	52	2	129	21	41	2	99	16
4	121	2	318	48	140	2	324	51	103	2	222	32
5	243	2	651	96	319	2	801	122	213	2	429	59
6	431	2	1170	171	627	2	1646	262	386	2	744	100
7	700	2	1917	279	1110	2	3093	496	637	2	1191	158
8	1065	2	2934	426	1820	2	5232	867	981	2	1794	236
9	1541	2	4263	618	2815	2	8455	1412	1433	2	2577	337
10	2143	2	5946	861	4159	2	12790	2186	2008	2	3564	464
11	2886	2	8025	1161	5922	2	18835	3238	2721	2	4779	620
12	3785	2	10542	1524	8180	2	26508	4635	3587	2	6246	808
13	4855	2	13539	1956	11015	2	36661	6438	4621	2	7989	1031
14	6111	2	17058	2463	14515	2	49054	8726	5838	2	10032	1292
15	7568	2	21141	3051	18774	2	64841	11572	7253	2	12399	1594
16	9241	2	25830	3726	23892	2	83576	15067	8881	2	15114	1940
17	11145	2	31167	4494	29975	2	106763	19296	10737	2	18201	2333
18	13295	2	37194	5361	37135	2	133702	24362	12836	2	21684	2776
19	15706	2	43953	6333	45490	2	166295	30362	15193	2	25587	3272
20	18393	2	51486	7416	55164	2	203540	37411	17823	2	29934	3824
21	21371	2	59835	8616	66287	2	247785	45618	20741	2	34749	4435

A.10 s4-s5-n

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	17	3	51	13	19	3	57	14	15	3	41	11
2	101	9	357	54	123	9	363	57	114	9	341	51
3	185	15	669	96	225	15	675	99	216	15	653	93
4	269	21	981	138	327	21	987	141	318	21	965	135
5	353	27	1293	180	429	27	1299	183	420	27	1277	177
6	437	33	1605	222	531	33	1611	225	522	33	1589	219
7	521	39	1917	264	633	39	1923	267	624	39	1901	261
8	605	45	2229	306	735	45	2235	309	726	45	2213	303
9	689	51	2541	348	837	51	2547	351	828	51	2525	345
10	773	57	2853	390	939	57	2859	393	930	57	2837	387
11	857	63	3165	432	1041	63	3171	435	1032	63	3149	429
12	941	69	3477	474	1143	69	3483	477	1134	69	3461	471
13	1025	75	3789	516	1245	75	3795	519	1236	75	3773	513
14	1109	81	4101	558	1347	81	4107	561	1338	81	4085	555
15	1193	87	4413	600	1449	87	4419	603	1440	87	4397	597
16	1277	93	4725	642	1551	93	4731	645	1542	93	4709	639
17	1361	99	5037	684	1653	99	5043	687	1644	99	5021	681
18	1445	105	5349	726	1755	105	5355	729	1746	105	5333	723
19	1529	111	5661	768	1857	111	5667	771	1848	111	5645	765
20	1613	117	5973	810	1959	117	5979	813	1950	117	5957	807
21	1697	123	6285	852	2061	123	6291	855	2052	123	6269	849

A.11 s4-s5-p

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	36	4	121	21	46	4	133	25	34	4	105	18
2	90	7	343	46	109	7	355	50	97	7	327	43
3	144	10	559	70	172	10	571	74	160	10	543	67
4	198	13	775	94	235	13	787	98	223	13	759	91
5	252	16	991	118	298	16	1003	122	286	16	975	115
6	306	19	1207	142	361	19	1219	146	349	19	1191	139
7	360	22	1423	166	424	22	1435	170	412	22	1407	163
8	414	25	1639	190	487	25	1651	194	475	25	1623	187
9	468	28	1855	214	550	28	1867	218	538	28	1839	211
10	522	31	2071	238	613	31	2083	242	601	31	2055	235
11	576	34	2287	262	676	34	2299	266	664	34	2271	259
12	630	37	2503	286	739	37	2515	290	727	37	2487	283
13	684	40	2719	310	802	40	2731	314	790	40	2703	307
14	738	43	2935	334	865	43	2947	338	853	43	2919	331
15	792	46	3151	358	928	46	3163	362	916	46	3135	355
16	846	49	3367	382	991	49	3379	386	979	49	3351	379
17	900	52	3583	406	1054	52	3595	410	1042	52	3567	403
18	954	55	3799	430	1117	55	3811	434	1105	55	3783	427
19	1008	58	4015	454	1180	58	4027	458	1168	58	3999	451
20	1062	61	4231	478	1243	61	4243	482	1231	61	4215	475
21	1116	64	4447	502	1306	64	4459	506	1294	64	4431	499

A.12 s4-t4p-n

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	148	4	347	50	150	4	376	58	118	4	269	37
2	300	4	719	98	312	4	862	132	242	4	565	73
3	452	4	1091	146	490	4	1428	222	366	4	861	109
4	604	4	1463	194	684	4	2074	328	490	4	1157	145
5	756	4	1835	242	894	4	2800	450	614	4	1453	181
6	908	4	2207	290	1120	4	3606	588	738	4	1749	217
7	1060	4	2579	338	1362	4	4492	742	862	4	2045	253
8	1212	4	2951	386	1620	4	5458	912	986	4	2341	289
9	1364	4	3323	434	1894	4	6504	1098	1110	4	2637	325
10	1516	4	3695	482	2184	4	7630	1300	1234	4	2933	361
11	1668	4	4067	530	2490	4	8836	1518	1358	4	3229	397
12	1820	4	4439	578	2812	4	10122	1752	1482	4	3525	433
13	1972	4	4811	626	3150	4	11488	2002	1606	4	3821	469
14	2124	4	5183	674	3504	4	12934	2268	1730	4	4117	505
15	2276	4	5555	722	3874	4	14460	2550	1854	4	4413	541
16	2428	4	5927	770	4260	4	16066	2848	1978	4	4709	577
17	2580	4	6299	818	4662	4	17752	3162	2102	4	5005	613
18	2732	4	6671	866	5080	4	19518	3492	2226	4	5301	649
19	2884	4	7043	914	5514	4	21364	3838	2350	4	5597	685
20	3036	4	7415	962	5964	4	23290	4200	2474	4	5893	721
21	3188	4	7787	1010	6430	4	25296	4578	2598	4	6189	757

A.13 s4-t4p-p

					SNF After AP				SNF After AP and Simp			
	Initial		SNF Only		After AP		After SNF		After AP		After SNF	
Id	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits	Size	Lits
1	93	4	219	36	89	4	221	37	68	4	161	26
2	169	4	405	60	166	4	434	68	130	4	309	44
3	245	4	591	84	245	4	667	103	192	4	457	62
4	321	4	777	108	328	4	920	142	254	4	605	80
5	397	4	963	132	415	4	1193	185	316	4	753	98
6	473	4	1149	156	506	4	1486	232	378	4	901	116
7	549	4	1335	180	601	4	1799	283	440	4	1049	134
8	625	4	1521	204	700	4	2132	338	502	4	1197	152
9	701	4	1707	228	803	4	2485	397	564	4	1345	170
10	777	4	1893	252	910	4	2858	460	626	4	1493	188
11	853	4	2079	276	1021	4	3251	527	688	4	1641	206
12	929	4	2265	300	1136	4	3664	598	750	4	1789	224
13	1005	4	2451	324	1255	4	4097	673	812	4	1937	242
14	1081	4	2637	348	1378	4	4550	752	874	4	2085	260
15	1157	4	2823	372	1505	4	5023	835	936	4	2233	278
16	1233	4	3009	396	1636	4	5516	922	998	4	2381	296
17	1309	4	3195	420	1771	4	6029	1013	1060	4	2529	314
18	1385	4	3381	444	1910	4	6562	1108	1122	4	2677	332
19	1461	4	3567	468	2053	4	7115	1207	1184	4	2825	350
20	1537	4	3753	492	2200	4	7688	1310	1246	4	2973	368
21	1613	4	3939	516	2351	4	8281	1417	1308	4	3121	386