From Default and Autoepistemic Logics to Disjunctive Answer Set Programs via the Logic of GK

Jiannin Ji[1] and **Hannes Strass**[2]

Abstract. The logic of knowledge and justified assumptions, also known as logic of grounded knowledge (GK), was proposed by Lin and Shoham as a general logic for nonmonotonic reasoning. To date, it has been used to embed in it default logic (propositional case), autoepistemic logic, Turner's logic of universal causation, and general logic programming under stable model semantics. Besides showing the generality of GK as a logic for nonmonotonic reasoning, these embeddings shed light on the relationships among these other logics. In this paper, for the first time, we show how the logic of GK can be embedded into disjunctive logic programming in a polynomial but non-modular translation with new variables. The result can then be used to compute the extension/expansion semantics of default logic, autoepistemic logic and Turner's logic of universal causation by disjunctive ASP solvers such as claspD(-2), DLV, GNT and cmodels.

1 Introduction

Lin and Shoham [21] proposed a logic with two modal operators $\mathbf{K}$ and $\mathbf{A}$, standing for knowledge and assumption, respectively. The idea is that one starts with a set of assumptions (those true under the modal operator $\mathbf{A}$), computes the minimal knowledge under this set of assumptions, and then checks to see if the assumptions were justified in that they agree with the resulting minimal knowledge. For instance, consider the GK formula $\mathbf{A}p \supset \mathbf{K}p$. If we assume p, then we can conclude that we know p, thus the assumption that p holds is justified, and we get a GK model where both $\mathbf{A}p$ and $\mathbf{K}p$ are true. (There is another GK model where we do not assume p and hence do not know p.) However, there is no GK model of $\neg\mathbf{A}p \supset \mathbf{K}p$: if we do not assume p, we are forced to conclude $\mathbf{K}p$, but then knowledge and assumptions do not coincide; if we do assume p, we cannot conclude that we know p and thus assuming p was not justified.

To date, there have been embeddings from default logic [30] and autoepistemic logic [27] to the logic of GK [21], from Turner's logic of universal causation [36] to the logic of GK [14], as well as from general logic programs [9] to the logic of GK [22]. Among other things, these embeddings shed new light on nonmonotonic reasoning, and have led to an interesting characterization of strong equivalence in logic programming [20, 22], and helped relate logic programming to circumscription [21] as the semantics of GK is just a minimization together with an identity checking after the minimization.

In this paper, for the first time, we consider computing models of GK theories by disjunctive logic programs. We shall propose a polynomial translation from a (pure) GK theory to a disjunctive logic program such that there is a one-to-one correspondence between GK models of the GK theory and answer sets of the resulting disjunctive logic program. The result can then be used to compute the extension/expansion semantics of default logic, autoepistemic logic and Turner's logic of universal causation by disjunctive ASP solvers such as claspD [7], claspD-2 [10], DLV [18], GNT [12] and cmodels [11]. To substantiate this claim, we have implemented the translation and report on some experiments that we conducted on the special case of computing extensions for Reiter's default logic [30]. The implementation, called gk2dlp, is available for download from the second author's home page at `http://informatik.uni-leipzig.de/~strass/gk2dlp/`.

Providing implementations for theoretical formalisms has a long tradition in nonmonotonic reasoning, for an overview see [6]. In fact, nonmonotonic reasoning itself originated from a desire to more accurately model the way humans reason, and was since its conception driven by applications in commonsense reasoning [25, 26]. Today, thanks to extensive research efforts, we know how closely interrelated the different formalisms for nonmonotonic reasoning are, and can use this knowledge to improve the scope of implementations.

This paper is organized as follows. Section 2 reviews logic programs, the logic of GK and default and autoepistemic logics. Section 3 presents our main result, the mapping from GK to disjunctive logic programming; due to space constraints, we could however not include any of the proofs. Section 4 presents our prototypical implementation, several experiments we conducted to analyze the translation, possible applications for it as well as a comparison with previous and related work. Section 5 concludes with ideas for future work.

2 Preliminaries

We assume a propositional language with two zero-place logical connectives $\top$ for tautology and $\bot$ for contradiction. We denote by $Atom$ the set of atoms, the signature of our language, and Lit the set of literals: $Lit = Atom \cup \{\neg p \mid p \in Atom\}$. A set I of literals is called *complete* if for each atom p, exactly one of $\{p, \neg p\}$ is in I.

In this paper, we identify an interpretation with a complete set of literals. If I is a complete set of literals, we use it as an interpretation when we say that it is a model of a formula, and we use it as a set of literals when we say that it entails a formula. In particular, we denote by $Th(I)$ the logical closure of I (considered to be a set of literals).

2.1 Logic Programming

A *nested expression* is built from literals using the 0-place connectives $\top$ and $\bot$, the unary connective "*not*" and the binary connectives "," and ";" for conjunction and disjunction. A *logic program* with nested expressions is a finite set of rules of the form $F \leftarrow G$, where F and G are nested expressions. The *answer set* of a logic

---

[1] School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

[2] Computer Science Institute, Leipzig University, Leipzig, Germany

program with nested expressions is defined as in [19]. Given a nested expression F and a set S of literals, we define when S satisfies F, written $S \models F$ below, recursively as follows (l is a literal):

- $S \models l$ if $l \in S$,
- $S \models \top$ and $S \not\models \bot$,
- $S \models \mathit{not}\, F$ if $S \not\models F$,
- $S \models F, G$ if $S \models F$ and $S \models G$, and
- $S \models F; G$ if $S \models F$ or $S \models G$.

S satisfies a rule $F \leftarrow G$ if $S \models F$ whenever $S \models G$. S satisfies a logic program P, written $S \models P$, if S satisfies all rules in P.

The *reduct* P^S of P related to S is the result of replacing every maximal subexpression of P that has the form $\mathit{not}\, F$ with $\bot$ if $S \models F$, and with $\top$ otherwise. For a logic program P without *not*, the *answer set* of P is any minimal consistent subset S of Lit that satisfies P. We use $\Gamma_P(S)$ to denote the set of answer sets of P^S. Now a consistent set S of literals is an *answer set* of P iff $S \in \Gamma_P(S)$. Every logic program with nested expressions can be equivalently translated to disjunctive logic programs with disjunctive rules of the form

$$l_1; \cdots ; l_k \leftarrow l_{k+1}, \ldots, l_t, \mathit{not}\, l_{t+1}, \ldots, \mathit{not}\, l_m, \\ \mathit{not}\, \mathit{not}\, l_{m+1}, \ldots, \mathit{not}\, \mathit{not}\, l_n$$

where $n \geq m \geq t \geq k \geq 0$ and $l_1, \ldots, l_n$ are propositional literals.

2.2 Default Logic

Default logic [30] is for closing gaps in incomplete knowledge bases. This is done by *defaults*, that allow to express rules of thumb such as "birds usually fly" and "tools usually work." For a given logical language, a *default* is any expression of the form $\phi : \psi_1, \ldots, \psi_n / \varphi$ where $\phi, \psi_1, \ldots, \psi_n, \varphi$ are formulas of the underlying language. A *default theory* is a pair (W, D), where W is a set of formulas and D is a set of defaults. The meaning of default theories is given through the notion of *extensions*. An extension of a default theory (W, D) is "interpreted as an acceptable set of beliefs that one may hold about the incompletely specified world W" [30]. For a default theory (W, D) and any set S of formulas let $\Gamma(S)$ be the smallest set satisfying (1) $W \subseteq \Gamma(S)$, (2) $Th(\Gamma(S)) = \Gamma(S)$, (3) If $\phi : \psi_1, \ldots, \psi_n / \varphi \in D$, $\phi \in \Gamma(S)$ and $\neg\psi_1, \ldots, \neg\psi_n \notin S$, then $\varphi \in \Gamma(S)$. A set E of formulas is called an *extension for* (W, D) iff $\Gamma(E) = E$.

2.3 Autoepistemic Logic

Moore [27] strives to formalize an ideally rational agent reasoning about its own beliefs. He uses a belief modality L to explicitly refer to the agent's belief within the language. Given a set A of formulas (the initial beliefs), a set T is an *expansion* of A if it coincides with the deductive closure of the set $A \cup \{L\varphi \mid \varphi \in T\} \cup \{\neg L\varphi \mid \varphi \notin T\}$. In words, T is an expansion if it equals what can be derived using the initial beliefs A and positive and negative introspection with respect to T itself. It was later discovered that this definition of expansions allows unfounded, self-justifying beliefs. Such beliefs are however not always desirable when representing the knowledge of agents.

2.4 The Logic of GK

The language of GK proposed by Lin and Shoham [21] is a modal propositional language with two modal operators, $\mathbf{K}$, for knowledge, and $\mathbf{A}$, for assumption. GK *formulas* are propositional formulas with $\mathbf{K}$ and $\mathbf{A}$. A GK *theory* is a set of GK formulas.

GK is a nonmonotonic logic, and its semantics is defined using the standard Kripke possible world interpretations. Informally speaking, a GK model is a Kripke interpretation where what is true under $\mathbf{K}$ is minimal and exactly the same as what is true under $\mathbf{A}$. The intuition here is that given a GK formula, one first makes some assumptions (those true under $\mathbf{A}$), then one minimizes the knowledge thus entailed, and finally checks to make sure that the initial assumption is justified in the sense that the minimal knowledge is the same as the initial assumption.

Formally, a *Kripke interpretation* M is a tuple $\langle W, \pi, R_K, R_A, s\rangle$, where W is a nonempty set of *possible worlds*, π a function that maps a possible world to an interpretation, R_K and R_A binary relations over W representing the accessibility relations for $\mathbf{K}$ and $\mathbf{A}$, respectively, and $s \in W$, called the *actual world* of M. The *satisfaction relation* $\models$ between a Kripke interpretation $M = \langle W, \pi, R_K, R_A, s\rangle$ and a GK formula F is defined in a standard way:

- $M \not\models \bot$,
- $M \models p$ iff $p \in \pi(s)$, where p is an atom,
- $M \models \neg F$ iff $M \not\models F$,
- $M \models F \wedge G$ iff $M \models F$ and $M \models G$,
- $M \models F \vee G$ iff $M \models F$ or $M \models G$,
- $M \models \mathbf{K}F$ iff $\langle W, \pi, R_K, R_A, \omega\rangle \models F$ for any $\omega \in W$, such that $(s, \omega) \in R_K$,
- $M \models \mathbf{A}F$ iff $\langle W, \pi, R_K, R_A, \omega\rangle \models F$ for any $\omega \in W$, such that $(s, \omega) \in R_A$.

Note that for any $\omega \in W$, $\pi(\omega)$ is an interpretation. We say that a Kripke interpretation M is a *model* of a GK formula F if M satisfies F, M is a *model* of a GK theory T if M satisfies every GK formula in T. In the following, given a Kripke interpretation M, we let

$$\mathbf{K}(M) = \{\, \phi \mid \phi \text{ is a propositional formula and } M \models \mathbf{K}\phi \,\},$$
$$\mathbf{A}(M) = \{\, \phi \mid \phi \text{ is a propositional formula and } M \models \mathbf{A}\phi \,\}.$$

Notice that $\mathbf{K}(M)$ and $\mathbf{A}(M)$ are always closed under classical logical entailment, that is, they are propositional theories.

Given a GK formula T, a Kripke interpretation M is a minimal model of T if M is a model of T and there does not exist another model M_1 of T such that $\mathbf{A}(M_1) = \mathbf{A}(M)$ and $\mathbf{K}(M_1) \subsetneq \mathbf{K}(M)$. We say that M is a *GK model* of T if M is a minimal model of T and $\mathbf{K}(M) = \mathbf{A}(M)$.

In this paper, we consider only GK formulas that do not contain nested occurrences of modal operators. Specifically, an $\mathbf{A}$-*atom* is a formula of the form $\mathbf{A}\phi$ and a $\mathbf{K}$-*atom* is a formula of the form $\mathbf{K}\phi$, where ϕ is a propositional formula. A GK formula is called a *pure GK formula* if it is formed from $\mathbf{A}$-atoms, $\mathbf{K}$-atoms and propositional connectives. Similarly, a *pure GK theory* is a set of pure GK formulas. Given a pure GK formula F, we denote

$$Atom_{\mathbf{K}}(F) = \{\, \phi \mid \mathbf{K}\phi \text{ is a } \mathbf{K}\text{-atom occurring in } F \,\},$$
$$Atom_{\mathbf{A}}(F) = \{\, \phi \mid \mathbf{A}\phi \text{ is an } \mathbf{A}\text{-atom occurring in } F \,\}.$$

For a pure GK theory T, we use $Atom_{\mathbf{K}}(T) = \bigcup_{F \in T} Atom_{\mathbf{K}}(F)$ and $Atom_{\mathbf{A}}(T) = \bigcup_{F \in T} Atom_{\mathbf{A}}(F)$ to denote their modal atoms.

So far, the applications of the logic of GK only ever use pure GK formulas. We now present some embeddings of well-known nonmonotonic knowledge representation languages into the logic of GK.

Default logic A (propositional) default theory $\Delta = (W, D)$ (under extension semantics) is translated into pure GK formulas in the following way: (1) Translate each $\phi \in W$ to $\mathbf{K}\phi$; (2) translate each $(\phi : \psi_1, \ldots, \psi_n / \varphi) \in D$ to $\mathbf{K}\phi \wedge \neg\mathbf{A}\neg\psi_1 \wedge \cdots \wedge \neg\mathbf{A}\neg\psi_n \supset \mathbf{K}\varphi$. For the weak extension semantics, a default $(\phi : \psi_1, \ldots, \psi_n / \varphi) \in D$ is translated to $\mathbf{A}\phi \wedge \neg\mathbf{A}\neg\psi_1 \wedge \cdots \wedge \neg\mathbf{A}\neg\psi_n \supset \mathbf{K}\varphi$.

Autoepistemic logic An L-sentence of autoepistemic logic that is in normal form [16], that is, a disjunction of the form $\neg L\phi \vee L\psi_1 \vee \cdots \vee L\psi_n \vee \varphi$, is (under expansion semantics) expressed as $\mathbf{A}\phi \wedge \neg\mathbf{A}\psi_1 \wedge \cdots \wedge \neg\mathbf{A}\psi_n \supset \mathbf{K}\varphi$. For strong expansion semantics, it becomes $\mathbf{K}\phi \wedge \neg\mathbf{A}\psi_1 \wedge \cdots \wedge \neg\mathbf{A}\psi_n \supset \mathbf{K}\varphi$.

Notice that the translation of default and autoepistemic theories into the logic of GK is compatible with Konolige's translation from default logic into autoepistemic logic [16]. Indeed, Konolige's translation perfectly aligns the weak extension semantics of default logic with expansion semantics for autoepistemic logic, and likewise for extension and strong expansion semantics.

Logic of universal causation A UCL formula F is translated to the pure logic of GK by replacing every occurrence of $\mathbf{C}$ by $\mathbf{K}$, adding $\mathbf{A}$ before each atom which is not in the range of $\mathbf{C}$ in F, and adding $\mathbf{A}p \vee \mathbf{A}\neg p$ for each atom p. For example, if F is $(p \wedge \neg q) \supset \mathbf{C}(p \wedge \neg q)$ and $Atom = \{p, q\}$, then the corresponding pure GK formula is $((\mathbf{A}p \wedge \neg\mathbf{A}q) \supset \mathbf{K}(p \wedge \neg q)) \wedge (\mathbf{A}p \vee \mathbf{A}\neg p) \wedge (\mathbf{A}q \vee \mathbf{A}\neg q)$.

Disjunctive logic programs Similarly, a disjunctive LP rule

$$p_1 \vee \cdots \vee p_k \leftarrow p_{k+1}, \ldots, p_l, not\, p_{l+1}, \ldots, not\, p_m,$$

where p's are atoms, corresponds to the following pure GK formula:

$$\mathbf{K}p_{k+1} \wedge \cdots \wedge \mathbf{K}p_l \wedge \neg\mathbf{A}p_{l+1} \wedge \cdots \wedge \neg\mathbf{A}p_m \supset \mathbf{K}p_1 \vee \cdots \vee \mathbf{K}p_k$$

3 Main Result: From Pure GK to Disjunctive ASP

Before presenting the translation, we introduce some notations. Let F be a pure GK formula, we use $tr_p(F)$ to denote the propositional formula obtained from F by replacing each occurrence of a $\mathbf{K}$-atom $\mathbf{K}\phi$ by k_ϕ and each occurrence of an $\mathbf{A}$-atom $\mathbf{A}\psi$ by a_ψ, where k_ϕ and a_ψ are new atoms with respect to ϕ and ψ respectively. For a pure GK theory T, we define $tr_p(T) = \bigwedge_{F \in T} tr_p(F)$. To illustrate these and the definitions that follow, we use a running example.

Example 1 (Normal Reiter default) *Consider the pure GK theory* $\{F\}$ *with* $F = \neg\mathbf{A}\neg p \supset \mathbf{K}p$ *corresponding to the default* $\top : p/p$. *Then* $tr_p(\{F\}) = \neg a_{\neg p} \supset k_p$, *where* $a_{\neg p}$ *and* k_p *are new atoms.*

Here we introduce a set of new atoms k_ϕ and a_ψ for each formula $\phi \in Atom_{\mathbf{K}}(T)$ and $\psi \in Atom_{\mathbf{A}}(T)$. Intuitively, the new atom k_ϕ (resp. a_ϕ) will be used to encode containment of the formula ϕ in $\mathbf{K}(M)$ (resp. $\mathbf{A}(M)$) of a GK model M for T.

Given a propositional formula ϕ and an atom a, we use ϕ^a to denote the propositional formula obtained from ϕ by replacing each occurrence of an atom p with a new atom p^a with respect to a. These formulas and new atoms will later be used in our main translation to perform the minimality check of the logic of GK's semantics.

We now stepwise work our way towards the main result. We start out with a result that relates a pure GK theory to a propositional formula that will later reappear in our main translation.

Proposition 1 *Let* T *be a pure GK theory. A Kripke interpretation* M *is a model of* T *if and only if there exists a model* I^* *of the propositional formula* $\Phi_T = tr_p(T) \wedge \varphi_{snd} \wedge \varphi^{\mathbf{K}}_{wit} \wedge \varphi^{\mathbf{A}}_{wit}$ *with*

$$\varphi_{snd} = \bigwedge_{\phi \in Atom_{\mathbf{K}}(T)} (k_\phi \supset \phi^k) \wedge \bigwedge_{\phi \in Atom_{\mathbf{A}}(T)} (a_\phi \supset \phi^a)$$

$$\varphi^{\mathbf{K}}_{wit} = \bigwedge_{\psi \in Atom_{\mathbf{K}}(T)} \left(\neg k_\psi \supset \left(\neg\psi^{k_\psi} \wedge \bigwedge_{\phi \in Atom_{\mathbf{K}}(T)} (k_\phi \supset \phi^{k_\psi}) \right) \right)$$

$$\varphi^{\mathbf{A}}_{wit} = \bigwedge_{\psi \in Atom_{\mathbf{A}}(T)} \left(\neg a_\psi \supset \left(\neg\psi^{a_\psi} \wedge \bigwedge_{\phi \in Atom_{\mathbf{A}}(T)} (a_\phi \supset \phi^{a_\psi}) \right) \right)$$

such that

- $\mathbf{K}(M) \cap Atom_{\mathbf{K}}(T) = \{\phi \mid \phi \in Atom_{\mathbf{K}}(T)$ *and* $I^* \models k_\phi\}$*;*
- $\mathbf{A}(M) \cap Atom_{\mathbf{A}}(T) = \{\phi \mid \phi \in Atom_{\mathbf{A}}(T)$ *and* $I^* \models a_\phi\}$.

The proposition examines the relationship between models of a pure GK theory and particular models of the propositional formula Φ_T. The first conjunct $tr_p(T)$ of the formula Φ_T indicates that the k-atoms and a-atoms in it can be interpreted in accordance with $\mathbf{K}(M)$ and $\mathbf{A}(M)$ such that $I^* \models tr_p(T)$ iff M is a model of T. The soundness formula φ_{snd} achieves that the sets $\{\phi \mid \phi \in Atom_{\mathbf{K}}(T)$ and $I^* \models k_\phi\}$ and $\{\phi \mid \phi \in Atom_{\mathbf{A}}(T)$ and $I^* \models a_\phi\}$ are consistent. The witness formulas φ_{wit} indicate that, if $I^* \models \neg k_\psi$ for some $\psi \in Atom_{\mathbf{K}}(T)$ (resp. $\psi \in Atom_{\mathbf{A}}(T)$) then there exists a model I' of $\mathbf{K}(M)$ (resp. $\mathbf{A}(M)$) such that $I' \models \neg\psi$, where I' is explicitly indicated by newly introduced p^{k_ψ} (resp. p^{a_ψ}) atoms. So intuitively, if a formula is not known (or not assumed), then there must be a witness for that. This condition is necessary: for instance, the set $\{k_p, k_q, \neg k_{p \wedge q}\}$ satisfies the formula $(k_{p \wedge q} \supset k_p) \wedge (k_{p \wedge q} \supset k_q)$, however, since $\mathbf{K}(M)$ is a theory there does not exist a Kripke interpretation M such that $p \in \mathbf{K}(M)$, $q \in \mathbf{K}(M)$ and $p \wedge q \notin \mathbf{K}(M)$.

Example 1 (Continued) *Formula* $\Phi_{\{F\}}$ *is given by:*

$$\begin{aligned} tr_p(F) &= \neg a_{\neg p} \supset k_p \\ \varphi_{snd} &= (k_p \supset p^k) \wedge (a_{\neg p} \supset \neg p^a) \\ \varphi^{\mathbf{K}}_{wit} &= \neg k_p \supset (\neg p^{k_p} \wedge (k_p \supset p^{k_p})) \\ \varphi^{\mathbf{A}}_{wit} &= \neg a_{\neg p} \supset (\neg\neg p^{a_{\neg p}} \wedge (a_{\neg p} \supset \neg p^{a_{\neg p}})) \end{aligned}$$

While Proposition 1 aligns Krikpe models and propositional models of the translation, there is yet no mention of GK's typical minimization step. This is the task of the next result, which extends the above relationship to GK models.

Proposition 2 *Let* T *be a pure GK theory. A Kripke interpretation* M *is a GK model of* T *if and only if there exists a model* I^* *of the propositional formula* Φ_T *such that*

- $\mathbf{K}(M) = \mathbf{A}(M) = Th\,(\{\phi \mid \phi \in Atom_{\mathbf{K}}(T)$ *and* $I^* \models k_\phi\}\,)$*;*
- *for each* $\psi \in Atom_{\mathbf{A}}(T)$*,*

  $$I^* \models a_\psi \textit{ iff } \psi \in Th(\{\phi \mid \phi \in Atom_{\mathbf{K}}(T) \textit{ and } I^* \models k_\phi\})$$

- *there does not exist another model* $I^{*\prime}$ *such that*
  - $I^{*\prime} \cap \{a_\phi \mid \phi \in Atom_{\mathbf{A}}(T)\} = I^* \cap \{a_\phi \mid \phi \in Atom_{\mathbf{A}}(T)\}$,
  - $I^{*\prime} \cap \{k_\phi \mid \phi \in Atom_{\mathbf{K}}(T)\} \subsetneq I^* \cap \{k_\phi \mid \phi \in Atom_{\mathbf{K}}(T)\}$.

Example 1 (Continued) *Clearly the intended reading of our running example is that there is no reason to assume that p is false, and the default lets us conclude that we know p. This is testified by the partial interpretation* $I^* = \{\neg a_{\neg p}, k_p, p^k, p^{a_{\neg p}}\}$ *where the remaining atoms are not relevant. It is easy to verify that* I^* *is a model for* $\Phi_{\{F\}}$ *and there is no model* $I^{*\prime}$ *with the properties above. Now* $k_p \in I^*$ *shows that p is known in the corresponding GK model.*

In Proposition 2, we only need to consider a Kripke interpretation M such that $\mathbf{A}(M) \cup \mathbf{K}(M)$ is consistent. This means that formula Φ_T can be modified to $\Psi_T = tr_p(T) \wedge \varphi'_{snd} \wedge \varphi'_{wit}$ with

$$\varphi'_{snd} = \bigwedge_{\phi \in Atom_{\mathbf{K}}(T)} (k_\phi \supset \phi) \wedge \bigwedge_{\phi \in Atom_{\mathbf{A}}(T)} (a_\phi \supset \phi)$$

$$\varphi'_{wit} = \bigwedge_{\psi \in Atom_{\mathbf{K}}(T)} \left(\neg k_\psi \supset \Phi^{\mathbf{K}}_\psi \right) \wedge \bigwedge_{\psi \in Atom_{\mathbf{A}}(T)} \left(\neg a_\psi \supset \Phi^{\mathbf{A}}_\psi \right)$$

$$\Phi_\psi^{\mathbf{K}} = \neg\psi^{k_\psi} \wedge \bigwedge_{\phi \in Atom_{\mathbf{K}}(T)} (k_\phi \supset \phi^{k_\psi}) \wedge \bigwedge_{\phi \in Atom_{\mathbf{A}}(T)} (a_\phi \supset \phi^{k_\psi})$$

$$\Phi_\psi^{\mathbf{A}} = \neg\psi^{a_\psi} \wedge \bigwedge_{\phi \in Atom_{\mathbf{K}}(T)} (k_\phi \supset \phi^{a_\psi}) \wedge \bigwedge_{\phi \in Atom_{\mathbf{A}}(T)} (a_\phi \supset \phi^{a_\psi})$$

So the soundness formula φ'_{snd} actually becomes easier, since soundness of knowledge and assumptions is enforced for one and the same vocabulary (the one from the original theory). The witness formulas become somewhat more complicated, as the witnesses have to respect both the knowledge as well as the assumptions of the theory. This is best explained by consulting our running example again.

Example 1 (Continued) *While F's propositionalization $tr_p(\{F\})$ stays the same, the soundness and witness formulas change in the step from formula $\Phi_{\{F\}}$ to formula $\Psi_{\{F\}}$. We only show the first conjunct of the witness formula φ'_{wit}, which is given by*

$$\neg k_p \supset \left(\neg p^{k_p} \wedge \left(k_p \supset p^{k_p}\right) \wedge \left(a_{\neg p} \supset \neg p^{k_p}\right)\right)$$

Intuitively, the formula expresses that whenever p is not known, then there must be a witness, that is, an interpretation where p is false. Since the witnessing interpretations could in principle be distinct for each $\mathbf{K}$-atom, they have to be indexed by the respective $\mathbf{K}$-atom they refer to, as in p^{k_p}. Of course, the witnesses have to obey all that is known and assumed, which is guaranteed in the last two conjuncts.

Using this new formula, the result of Proposition 2 can be restated.

Proposition 3 *Let T be a pure GK theory. A Kripke interpretation M is a GK model of T if and only if there exists a model I^* of the propositional formula Ψ_T such that*

- $\mathbf{K}(M) = \mathbf{A}(M) = \mathit{Th}\,(\{\phi \mid \phi \in Atom_{\mathbf{K}}(T) \text{ and } I^* \models k_\phi\})$*;*
- *for each $\psi \in Atom_{\mathbf{A}}(T)$,*

  if $I^* \models a_\psi$ then $\psi \in \mathit{Th}(\{\phi \mid \phi \in Atom_{\mathbf{K}}(T) \text{ and } I^* \models k_\phi\})$
- *there does not exist another model $I^{*\prime}$ of Φ_T such that*
  - $I^{*\prime} \cap \{a_\phi \mid \phi \in Atom_{\mathbf{A}}(T)\} = I^* \cap \{a_\phi \mid \phi \in Atom_{\mathbf{A}}(T)\}$,
  - $I^{*\prime} \cap \{k_\phi \mid \phi \in Atom_{\mathbf{K}}(T)\} \subsetneq I^* \cap \{k_\phi \mid \phi \in Atom_{\mathbf{K}}(T)\}$.

We are now ready for our main result, translating a pure GK theory to a disjunctive logic program. First, we introduce some notations. Let T be a pure GK theory, we use $tr_{ne}(T)$ to denote the nested expression obtained from Ψ_T by first converting it to negation normal form[3], then replacing "$\wedge$" by "," and "$\vee$" by ";". A propositional formula ϕ can be equivalently translated to conjunctive normal form (involving at most linear blowup)

$$\begin{aligned}(p_1 \vee \cdots \vee p_t \vee \neg p_{t+1} \vee \cdots \vee \neg p_m) \wedge \dots \\ \wedge\, (q_1 \vee \cdots \vee q_k \vee \neg q_{k+1} \vee \cdots \vee \neg q_n)\end{aligned}$$

where p's and q's are atoms; we use $tr_c(\phi)$ to denote the set of rules

$$p_1; \dots; p_t \leftarrow p_{t+1}, \dots, p_m \qquad \dots \qquad q_1; \dots; q_k \leftarrow q_{k+1}, \dots, q_n$$

We use $\widehat{\phi}$ to denote the propositional formula obtained from ϕ by replacing each occurrence of an atom p by a new atom $\hat{p}$.

We use T^* to denote the propositional formula obtained from the formula Φ_T by replacing each occurrence of an atom p (except atoms in $\{a_\phi \mid \phi \in Atom_{\mathbf{A}}(T)\}$) by a new atom p^*. Intuitively, each atom that is not an a-atom is replaced by a new atom.

Notice that $tr_{ne}(T)$ is obtained from Ψ_T while T^* is obtained from Φ_T. Intuitively, by Proposition 3, $tr_{ne}(T)$ is used to restrict interpretations for introduced k-atoms and a-atoms so that these interpretations serve as candidates for GK models, and by Proposition 1, T^* constructs possible models of the GK theory which are later used to test whether these models prevent the candidate to be a GK model.

Inspired by the linear translation from parallel circumscription into disjunctive logic programs in [13], we have the following theorem.

Theorem 1 *Let T be a pure GK theory. A Kripke interpretation M is a GK model of T if and only if there exists an answer set S of the logic program $tr_{lp}(T)$:*

$$\begin{array}{lll}
(1) & \bot \leftarrow \mathit{not}\; tr_{ne}(T) & \\
(2) & p'; \neg p' \leftarrow \top & (\text{for each atom } p' \text{ occurring in } tr_{ne}(T)) \\
(3) & u; A \leftarrow B & (\text{for each rule } A \leftarrow B \text{ in } tr_c(T^*)) \\
(4) & u; c_{\phi_1}; \cdots; c_{\phi_m} \leftarrow \top & (\{\phi_1, \dots, \phi_m\} = Atom_{\mathbf{K}}(T)) \\
(5) & u \leftarrow c_\phi, \mathit{not}\; k_\phi & (\text{for each } \phi \in Atom_{\mathbf{K}}(T)) \\
(6) & u \leftarrow k^*_\phi, \mathit{not}\; k_\phi & (\text{for each } \phi \in Atom_{\mathbf{K}}(T)) \\
(7) & u \leftarrow c_\phi, k^*_\phi, \mathit{not}\; \neg k_\phi & (\text{for each } \phi \in Atom_{\mathbf{K}}(T)) \\
(8) & u; c_\phi; k^*_\phi \leftarrow \mathit{not}\; \neg k_\phi & (\text{for each } \phi \in Atom_{\mathbf{K}}(T)) \\
(9) & p^* \leftarrow u & (\text{for each new atom } p^* \text{ in } tr_c(T^*)) \\
(10) & c_\phi \leftarrow u & (\text{for each } \phi \in Atom_{\mathbf{K}}(T)) \\
(11) & \bot \leftarrow \mathit{not}\; u & \\
(12) & v; A \leftarrow B &
\end{array}$$

(for each rule $A \leftarrow B$ in the $tr_c(\cdot)$ translation of

$$\bigwedge_{\phi \in Atom_{\mathbf{K}}(T)} (k_\phi \supset \widehat{\phi}) \wedge \neg \bigwedge_{\phi \in Atom_{\mathbf{A}}(T)} (a_\phi \supset \widehat{\phi}))$$

$$(13) \qquad \hat{p} \leftarrow v$$

(for each atom $\hat{p}$ except k-atoms and a-atoms in $tr_c(\cdot)$ of

$$\bigwedge_{\phi \in Atom_{\mathbf{K}}(T)} (k_\phi \supset \widehat{\phi}) \wedge \neg \bigwedge_{\phi \in Atom_{\mathbf{A}}(T)} (a_\phi \supset \widehat{\phi}))$$

$$(14) \qquad \bot \leftarrow \mathit{not}\; v$$

where u, v, and c_ϕ (for each $\phi \in Atom_{\mathbf{K}}(T)$) are new atoms, such that $\mathbf{K}(M) = \mathbf{A}(M) = \mathit{Th}(\{\phi \mid \phi \in Atom_{\mathbf{K}}(T) \text{ and } k_\phi \in S\})$.

The intuition behind the construction is as follows:

- (1) and (2) in $tr_{lp}(T)$: I^* is a model of the formula Ψ_T.
- (3) to (8): if there exists a model $I^{*\prime}$ of the formula Φ_T with
  - $I^* \cap \{a_\phi \mid \phi \in Atom_{\mathbf{A}}(T)\} = I^{*\prime} \cap \{a_\phi \mid \phi \in Atom_{\mathbf{A}}(T)\}$
  - $I^{*\prime} \cap \{k_\phi \mid \phi \in Atom_{\mathbf{K}}(T)\} \subsetneq I^* \cap \{k_\phi \mid \phi \in Atom_{\mathbf{K}}(T)\}$,

  then there exists a set S^* constructed from new atoms in $tr_c(T^*)$ (which is a copy of the formula Φ_T with same a_ϕ for each $\phi \in Atom_{\mathbf{A}}(T)$) and c_ϕ for some $\phi \in Atom_{\mathbf{K}}(T)$ such that S^* satisfies rules (3) to (8) and $u \notin S^*$.
- (9) and (10): if there is such a set S^* then it is the least set containing u, all p^*'s and c-atoms.
- (11): such a set S^* should not exist. (See item 3 in Proposition 3.)
- (12) and (13): if there exists a model of the propositional formula $\bigwedge_{\phi \in Atom_{\mathbf{K}}(T)}(k_\phi \supset \widehat{\phi}) \wedge \neg \bigwedge_{\phi \in Atom_{\mathbf{A}}(T)}(a_\phi \supset \widehat{\phi})$, then v should not occur in the minimal model of the program.

---

[3] A propositional formula is in Negation Normal Form (NNF) if negation occurs only immediately above atoms, and $\{\bot, \top, \neg, \wedge, \vee\}$ are the only allowed connectives.

- (14): $\bigwedge_{\phi\in Atom_{\mathbf{K}}(T)}(k_\phi \supset \widehat{\phi}) \wedge \neg \bigwedge_{\phi\in Atom_{\mathbf{A}}(T)}(a_\phi \supset \widehat{\phi})$ should not be consistent. (This is necessary by item 2 in Proposition 3.)

Example 1 (Continued) *For our running example theory* $\{F\}$ *with* $F = \neg\mathbf{A}\neg p \supset \mathbf{K}p$*, we find that the logic program translation* $tr_{lp}(\{F\})$ *has a single answer set* S *with* $k_p \in S$*. Thus by Theorem 1 we can conclude that the GK theory* $\{F\}$ *has a single GK model* M *in which* $\mathbf{K}(M) = \mathit{Th}(\{p\})$.

Computational complexity We have seen in Section 2.4 that disjunctive logic programs can be modularly and equivalently translated into pure formulas of the logic of GK. Conversely, Theorem 1 shows that pure GK formulas can be equivalently translated into disjunctive logic programs. Eiter and Gottlob showed that the problem of deciding whether a disjunctive logic program has an answer set is Σ_2^P-complete [8]. In combination, these results yield the following straightforward complexity result for the satisfiability of pure GK.

Proposition 4 *Let* T *be a pure GK theory. The problem of deciding whether* T *has a GK model is* Σ_2^P*-complete.*

4 Implementation

We have implemented the translation of Theorem 1 into a working prototype gk2dlp. The program is written in Prolog and uses the disjunctive ASP solver claspD-2 [10], which was ranked first place in the 2013 ASP competition (`http://www.mat.unical.it/ianni/storage/aspcomp-2013-lpnmrtalk.pdf`). When computing answer sets of the translated logic programs, we use claspD-2's "`--project`" option to project out all atoms but $\mathbf{K}$-atoms, since these suffice to reconstruct GK models.

Our prototype is the first implementation of the (pure) logic of GK to date. The restriction to pure formulas seems harmless since all known applications of the logic of GK use only pure formulas. We remark that gk2dlp implements default and autoepistemic logics such that input and target language are of the same complexity.

Evaluation To have a scalable problem domain and inspired by dl2asp [4], we chose the fair division problem [2] for experimental evaluation. An instance of the fair division problem consists of a set of agents, a set of goods, and for each agent a set of constraints that intuitively express which sets of goods the agent is willing to accept. A solution is then an assignment of goods to agents that is a partition of all goods and satisfies all agents' constraints. Bouveret & Lang [2] showed that the problem can be naturally encoded in default logic.

We created random instances of the fair division problem with increasing numbers of agents and goods. We then applied the translation of [2], furthermore the translation from default logic into the logic of GK, then invoked gk2dlp to produce logic programs and finally used gringo 3.0.3 and claspD version 2 (revision 6814) to compute all answer sets of these programs, thus all extensions of the original default theory corresponding to all solutions of the problem instance. The experiments were conducted on a Lenovo laptop with an Intel Core i3 processor with 4 cores and 4GB of RAM running Ubuntu 12.04. We recorded the size of the default theory, the size of the translated logic program, the translation time and the solving time, as well as the number of solutions obtained. We started out with 2 agents and 2 goods, and stepwise increased these numbers towards 6. For each combination in $(a, g) \in \{2, \dots, 6\} \times \{2, \dots, 6\}$, we tested 20 randomly generated instances. Random generation here means that we create agents' preferences by iteratively drawing random subsets of goods to add to an agent's acceptable subsets with probability P, where P is initialized with 1 and discounted by the factor $\frac{g-1}{g}$ for each subset that has been drawn.

Below we show two scatter plots, where each point represents a single problem instance. The first plot depicts the size increase of the translation of this paper. The size of the GK formula of a given default theory is roughly the same as that of the default theory. As the plot shows, the increase in size from default theory to logic program is polynomial, albeit with a low exponent. The second plot shows the solving time in relation to the size of the default theory, where the time axis is logarithmic. We can see that the runtime behavior of gk2dlp is satisfactory, although not competitive with that of dl2asp [4]. However, a direct comparison of the two systems is problematic since [4] do not describe how they create random instances of the fair division problem, and more importantly dl2asp is especially engineered for default logic, while gk2dlp is a more general system.

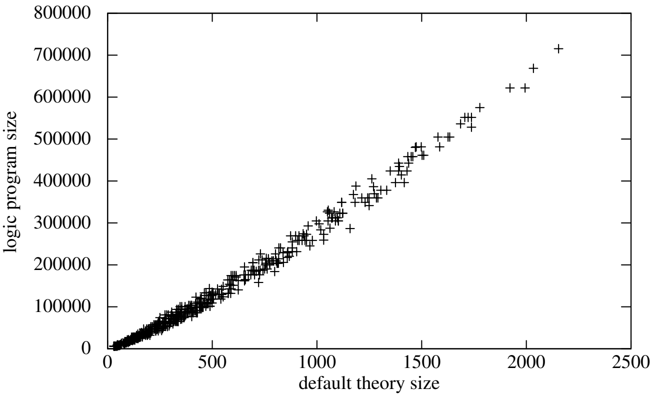

Figure 1. Size increase from default theory to disjunctive logic program. The size of a default theory is the sum of sizes of all formulas occurring in the default theory; the size of a formula is the number of atoms plus the number of connectives. The size of a logic program is the sum of sizes of all rules; the size of a rule is the number of literals in it.

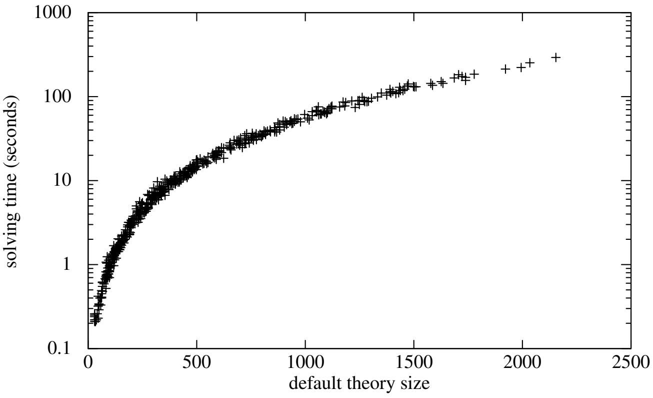

Figure 2. Solving time (log scale) with respect to default theory size.

Applications We see immediate applicability of the translation of the present paper to several areas. Reiter [31] provided a theory of diagnosis from first principles, and showed how default logic can be used as an implementation device. Cadoli, Eiter and Gottlob [3] proposed to use default logic as an expressive query language on top of relational databases, and gave an example of achieving strategic behavior in an economic setting. In reasoning about actions, Thielscher [35] used default logic to solve the qualification problem of dealing with unexpected action failures. Martin and Thielscher [24] later provided an implementation of that approach where extensions are enumerated in Prolog. Recently, Baumann et al. [1] introduced a method for default reasoning in action theories, that is, an approach to the question what normally holds in a dynamic domain. Our translation yields an implementation of their approach, something that they stated as future work and later achieved to a limited extent (for a restricted sublanguage of their framework [34]).

Related work There are few approaches that implement as broad a range of propositional nonmonotonic knowledge representation languages as gk2dlp. Two notable exceptions are the works of Junker and Konolige [15], who implemented both autoepistemic and default logics by translating them to truth maintenance systems; and Niemelä [29], who provides a decision procedure for autoepistemic logic which also incorporates extension semantics for default logics. Other approaches are restricted to specific languages. For default logic, the recent system dl2asp [4] translates default theories to normal (non-disjunctive) logic programs; the translation figures out all implication relations between formulas occurring in the default theory, just as [15] did. The authors of dl2asp [4] already observed that default logic and disjunctive logic programs are of the same complexity; they even stated the search for a polynomial translation from the former to the latter (that we achieved in this paper) as future work. Gadel [28] uses a genetic algorithm to compute extensions of a default theory; likewise the system DeReS [5] is not translation-based but directly searches for extensions; similarly the XRay system [33] natively implements local query-answering in default logics. Risch and Schwind [32] describe a tableaux-based algorithm for computing all extensions of general default theories, but do not report runtimes for their Prolog-based implementation. For autoepistemic logic, Marek and Truszczyński [23] investigate sceptical reasoning with respect to Moore's expansion semantics.

5 Discussion

We have presented the first translation of pure formulas of the logic of GK to disjunctive answer set programming. Among other things, this directly leads to implementations of Turner's logic of universal causation as well as implementations of default and autoepistemic logics under different semantics. We have prototypically implemented the translation and experimentally analysed its performance, which we found to be satisfactory given the system's generality.

In the future, we plan to integrate further nonmonotonic reasoning formalisms. This is more or less straightforward due to the generality of this work: to implement a language, it suffices to provide a translation into pure formulas of GK, then Theorem 1 of this paper does the rest. A particular formalism we want to look at is Lakemeyer and Levesque's logic of only-knowing [17]. We also plan to study the approaches mentioned as applications in the previous section to try out our translation and implementation on agent-oriented AI problems.

References

[1] Ringo Baumann, Gerhard Brewka, Hannes Strass, Michael Thielscher, and Vadim Zaslawski, 'State Defaults and Ramifications in the Unifying Action Calculus', in *KR*, pp. 435–444, (2010).

[2] Sylvain Bouveret and Jérôme Lang, 'Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity.', *JAIR*, **32**, 525–564, (2008).

[3] Marco Cadoli, Thomas Eiter, and Georg Gottlob, 'Default logic as a query language', in *KR*, pp. 99–108, (1994).

[4] Yin Chen, Hai Wan, Yan Zhang, and Yi Zhou, 'dl2asp: Implementing Default Logic via Answer Set Programming', in *JELIA*, volume 6341, pp. 104–116, (2010).

[5] Pawel Cholewiński, Victor W. Marek, Miroslaw Truszczyński, and Artur Mikitiuk, 'Computing with default logic', *AIJ*, **112**(1), 105–146, (1999).

[6] Jürgen Dix, Ulrich Furbach, and Ilkka Niemelä, 'Nonmonotonic reasoning: Towards efficient calculi and implementations', *Handbook of Automated Reasoning*, **2**(18), 1121–1234, (2001).

[7] Christian Drescher, Martin Gebser, Torsten Grote, Benjamin Kaufmann, Arne König, Max Ostrowski, and Torsten Schaub, 'Conflict-Driven Disjunctive Answer Set Solving', in *KR*, pp. 422–432, (2008).

[8] Thomas Eiter and Georg Gottlob, 'On the computational cost of disjunctive logic programming: Propositional case', *AMAI*, **15**(3–4), 289–323, (1995).

[9] Paolo Ferraris, 'Answer sets for propositional theories', in *LPNMR*, pp. 119–131, (2005).

[10] Martin Gebser, Benjamin Kaufmann, and Torsten Schaub, 'Advanced conflict-driven disjunctive answer set solving', in *IJCAI*, (2013).

[11] Enrico Giunchiglia, Yuliya Lierler, and Marco Maratea, 'Answer Set Programming Based on Propositional Satisfiability', *J. Autom. Reasoning*, **36**(4), 345–377, (2006).

[12] Tomi Janhunen and Ilkka Niemelä, 'GnT – A Solver for Disjunctive Logic Programs', in *LPNMR*, pp. 331–335, Berlin, Heidelberg, (2004).

[13] Tomi Janhunen and Emilia Oikarinen, 'Capturing parallel circumscription with disjunctive logic programs', in *Logics in Artificial Intelligence*, 134–146, (2004).

[14] Jianmin Ji and Fangzhen Lin, 'From Turner's Logic of Universal Causation to the Logic of GK', in *Correct Reasoning*, volume 7265, pp. 380–385, (2012).

[15] Ulrich Junker and Kurt Konolige, 'Computing the Extensions of Autoepistemic and Default Logics with a Truth Maintenance System', in *AAAI*, pp. 278–283, (1990).

[16] Kurt Konolige, 'On the Relation Between Default and Autoepistemic Logic', *AIJ*, **35**(3), 343–382, (1988).

[17] Gerhard Lakemeyer and Hector J. Levesque, 'Only-knowing: Taking it beyond autoepistemic reasoning', in *AAAI*, pp. 633–638, (2005).

[18] Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello, 'The DLV system for knowledge representation and reasoning', *ACM Transactions on Computational Logic*, **7**(3), 499–562, (2006).

[19] Vladimir Lifschitz, Lappoon R. Tang, and Hudson Turner, 'Nested expressions in logic programs', *AMAI*, **25**(3-4), 369–389, (1999).

[20] Fangzhen Lin, 'Reducing strong equivalence of logic programs to entailment in classical propositional logic', in *KR*, pp. 170–176, (2002).

[21] Fangzhen Lin and Yoav Shoham, 'A logic of knowledge and justified assumptions', *AIJ*, **57**(2-3), 271–289, (1992).

[22] Fangzhen Lin and Yi Zhou, 'From answer set logic programming to circumscription via logic of GK', *AIJ*, **175(1)**, 264–277, (2011).

[23] V. Wiktor Marek and Miroslaw Truszczyński, 'Computing intersection of autoepistemic expansions', in *LPNMR*, pp. 37–50, (1991).

[24] Yves Martin and Michael Thielscher, 'Addressing the Qualification Problem in FLUX', in *KI/ÖGAI*, pp. 290–304, (2001).

[25] John McCarthy, 'Circumscription – a form of non-monotonic reasoning', *AIJ*, **13**, 295–323, (1980).

[26] John McCarthy, 'Applications of circumscription to formalizing commonsense knowledge', *AIJ*, **28**, 89–118, (1986).

[27] Robert Moore, 'Semantical considerations on nonmonotonic logic', *AIJ*, **25(1)**, 75–94, (1985).

[28] Pascal Nicolas, Frédéric Saubion, and Igor Stéphan, 'Gadel: a genetic algorithm to compute default logic extensions', in *ECAI*, pp. 484–490, (2000).

[29] Ilkka Niemelä, 'A decision method for nonmonotonic reasoning based on autoepistemic reasoning', *J. Autom. Reasoning*, **14**(1), 3–42, (1995).

[30] Raymond Reiter, 'A logic for default reasoning', *AIJ*, **13**(1-2), 81–132, (1980).

[31] Raymond Reiter, 'A theory of diagnosis from first principles', *AIJ*, **32**(1), 57–95, (1987).

[32] Vincent Risch and Camilla Schwind, 'Tableaux-based characterization and theorem proving for default logic', *J. Autom. Reasoning*, **13**(2), 223–242, (1994).

[33] Torsten Schaub and Pascal Nicolas, 'An implementation platform for query-answering in default logics: The XRay system, its implementation and evaluation', in *LPNMR*, 441–452, (1997).

[34] Hannes Strass, 'The draculasp system: Default reasoning about actions and change using logic and answer set programming', in *NMR*, (2012).

[35] Michael Thielscher, 'Causality and the Qualification Problem', in *KR*, pp. 51–62, (1996).

[36] Hudson Turner, 'Logic of universal causation', *AIJ*, **113**(1), 87–123, (1999).