Experimental Comparative Study of Compilation-Based Inference in Bayesian and Possibilitic Networks

Raouia Ayachi[1,2], Nahla Ben Amor[1], and Salem Benferhat[2]

[1] LARODEC Laboratory, ISG, University of Tunis, Tunisia, 2000
[2] CRIL-CNRS, University of Artois, France, 62307

Abstract. Graphical models are important tools for representing and analyzing uncertain information. Diverse inference methods were developed for efficient computations in these models. In particular, compilation-based inference has recently triggered much research, especially in the probabilistic and the possibilistic frameworks. Even though the inference process follows the same principle in the two frameworks, it depends strongly on the specificity of each of them, namely in the interpretation of handled values (probability\possibility) and appropriate operators (*\min and +\max). This paper emphasizes on common points and unveils differences between the compilation-based inference process in the probabilistic and the possibilistic setting from a spatial viewpoint.

Keywords: Bayesian networks, Qualitative graphical models, Possibility theory, Compilation-based inference.

1 Introduction

Graphical models are a powerful family of models for representing and analyzing uncertain information. They are characterized by their explicitness and clarity. *Bayesian networks* are studied under the broader class of probabilistic graphical models. However, the probability theory in such models is only appropriate when all numerical data are available, which is not always possible. Several non-classical theories of uncertainty have been proposed in order to deal with uncertain and imprecise data. We are in particular interested in *possibility theory* [8,9]. The last decade has seen a virtual explosion of applications of propositional logic. One emerging application is *knowledge compilation.* It consists in preprocessing the propositional theory only once in an off-line phase, with the goal of making frequent on-line queries efficient [2]. One of the most prominent successful applications of knowledge compilation is in the context of graphical models. In fact, in [3], authors focused on compiling Bayesian networks using DNNFs. In [1], we studied the possibilistic adaptation of some compilation-based inference methods using Π-DNNFs. The objective behind these methods is to ensure an efficient computation of a-posteriori probability or possibility

A. Petrosino et al. (Eds.): WILF 2011, LNAI 6857, pp. 155–163, 2011.

degrees given some evidence on some variables. We intend that is considerable to accomplish a comparison study in which we unveil the most compact framework in this context. In this paper, we investigate the extent to which possibility theory can be used to reduce sizes of compiled bases. In fact, we will compare the probabilistic and the possibilistic approaches and prove the importance of the possibilistic setting versus the probabilistic setting. The remaining paper is organized as follows: Section 2 presents a brief refresher on possibility theory and compilation. Section 3 describes both the probabilistic and the possibilistic approaches by focusing on similarities and differences between the two settings. Experimental study is presented in Section 4. Section 5 concludes the paper.

2 Basic Concepts

Let $V = \{X_1, X_2, ..., X_N\}$ be a set of variables. We denote by D_{X_i} the domain associated with the variable X_i. By x_i (resp. x_{ij}), we denote any of the instances of X_i (resp. the j^{th} instance of X_i). When there is no confusion we use x_i to mean any instance of X_i. In the n-ary case, $D_{X_i} = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ where n is the number of instances of X_i. By v we denote instantiations of all variables $X_i \in V$. Ω denotes the universe of discourse, which is the cartesian product of all variable domains in V. Each element $\omega \in \Omega$ is called an *interpretation*, a possible *world* or a *state* of Ω. $\omega[X_i] = x_i$ denotes an instantiation of X_i in ω.

In this paper, we are interested in two uncertainty frameworks, namely the standard one, i.e., the probabilistic setting and the non-standard possibility theory [8,9]. The basic building block in this theory is the concept of *possibility distribution* π, which is a mapping from the universe of discourse Ω to the unit interval [0, 1] such that $\pi(\omega) = 1$ means that the realization of ω is totally possible and $\pi(\omega) = 0$ means that ω is an impossible state. In the extreme case of total ignorance, $\pi(\omega) = 1$, $\forall \omega \in \Omega$. It is generally assumed that there exists at least a state ω which is totally possible. In this case, π is said to be normalized. From π, we can compute two dual measures $\Pi(\phi) = max_{\omega \in \phi} \pi(\omega)$ and $N(\phi) = 1 - \Pi(\neg\phi)$ evaluating respectively to which extent ϕ is consistent with the knowledge represented by π and to which level ϕ is certainly implied by this knowledge. Contrarily to the probabilistic case where $P(\neg\phi) = 1 - P(\phi)$, possibility and necessity measures are weakly linked. In possibility theory, conditioning is defined by the following counterpart of the Bayesian rule: $\forall \omega,\ \pi(\omega) = \min(\pi(\omega \mid \psi), \Pi(\psi))$. $\pi(\omega \mid \psi)$ and $\Pi(\psi)$ are combined using a min operation, according to the *ordinal* interpretation of the possibilistic scale[1]. In what follows, we will use some generic notations, i.e., the conjunctive operator $\otimes$ corresponding to $\prod$ and min, the disjunctive operator $\oplus$ corresponding to $\sum$ and max, in the probabilistic and the possibilistic case, resp. $\propto$ denotes the probability or the possibility degree depending on the setting. $\oslash = \{B, \Pi\}$, i.e., if we use $\oslash = B$ we mean the probabilistic setting and if use $\oslash = \Pi$ we mean the possibilistic case.

[1] The *numerical* interpretation of possibility theory uses the product instead of the min, but this is out the scope of the present study.

2.1 Bayesian and Possibilistic Networks

Bayesian and min-based graphical models (denoted by G^B and G^Π, resp.) share the same graphical component, i.e., a DAG where nodes represent variables in V and edges encode different (in)dependence relationships. The major difference resides in their numerical component since in G^B different links are quantified via probability distributions while in G^Π, we use possibility distributions. Formally, each node X_i in a G^B (resp. G^Π) is represented by a local normalized probability (resp. possibility) distribution in the context of its parents, denoted by $U_i = \{U_{i1}, U_{i2}, ..., U_{im}\}$ where m is the number of parents of X_i. In what follows, we use x_i, u_i, u_{ij} to denote, resp. possible instances of X_i, U_i and U_{ij}.

The set of a priori and conditional probability (resp. possibility) distributions induces a unique joint distribution via a chain rule based on the product in the probabilistic setting and min in the possibilistic setting [11].

2.2 Compilation

A logical form qualifies as a *target compilation language* if it supports some set of nontrivial *transformations* and *queries* in polynomial time with respect to the size of compiled bases [7]. We will review in this section the target compilation languages relevant to the present paper. The *Decomposable Negation Normal Form (DNNF)* is a universal language presenting a number of properties that makes it tractable and of a great interest. DNNF, which is qualified as succinct, is an *Negation Normal Form (NNF)* language satisfying the *decomposability* property stating that: conjuncts of any conjunction do not share variables [4]. A set of important properties may be imposed to DNNF, for instance, *determinism* and *smoothness* giving rise to the sd-DNNF. The DNNF compilation language (or one of its variants) supports a rich set of polynomial-time logical operations [7]. For queries, within the most common queries, we cite *model counting*. For transformations, we focus on *conditioning*, *forgetting* and *minimization*:

- *Conditioning*: Let α be a propositional formula. Let ρ be a consistent term, then conditioning α on ρ, denoted by $\alpha|\rho$ generates a new formula where each variable P_i of α is replaced by $\top$ if P_i is consistent with ρ, and by $\bot$ otherwise.
- *Forgetting*: Let α be a propositional formula, let P be a finite set of propositional variables P_i, then the forgetting of P from α, denoted by $\exists P.\alpha$ is a formula that does not mention any variable P_i from P.
- *Minimization*: Let α be a propositional formula, then the minimization of α is a formula β such that the cardinality of all β's models is equal to the minimum cardinality models of α [4]. Recall that the cardinality of a model corresponds to the number of variables set to True ($\top$) or False ($\bot$).

Π-DNNF [1] is a possibilistic version of DNNF in which conjunctions and disjunctions are substituted by minimum and maximum operators, respectively. It is considered as a special case of VNNFs [10].

3 DNNF vs *Π*-DNNF

It was shown recently that compiling Bayesian networks corresponds to factoring multi-linear functions [5]. In the possibilistic framework, we have shown that compiling possibilistic networks corresponds to factoring possibilistic functions [1]. In this section, we will propose a generic approach handling both probabilistic and possibilistic settings to raise awareness about these approaches and reveal the differences between them.

3.1 From Encoding to Inference

The multi-linear (resp. possibilistic) function $f^{\oslash}$ of a network $G^{\oslash}$ contains two types of propositional variables. An *indicator variable* λ_{x_i} is associated for each value x_i of each X_i in $G^{\oslash}$. Furthermore, a *parameter variable* $\theta_{x_i|u_i}$ is associated for each network parameter $\propto (x_i|u_i)$ in $G^{\oslash}$. Equation (1) expresses the generic function $f^{\oslash}$ s.t. $\oplus$ denotes maximum or probabilistic sum and $\otimes$ denotes minimum or product depending on the setting.

$$f^{\oslash} = \bigoplus_{v} \bigotimes_{(x_i,u_i)\sim v} \lambda_{x_i}\theta_{x_i|u_i} \tag{1}$$

To compute the probability or the possibility of an evidence e, $f^{\oslash}$ should be evaluated after setting appropriate values to indicator variables depending on e. In both settings, $f^{\oslash}$ has an exponential size, so it should be interesting to encode such function using a propositional theory to represent it more compactly. The CNF propositional language is chosen since it is often convenient for compactly encoding knowledge bases [7].

Definition 1. *Let $G^{\oslash}$ be a network ($\oslash = \{B, \Pi\}$), $\lambda_{x_{ij}}$, $(i = 1, \ldots, N)$,$(j = \{1, \ldots, n\})$ be the set of evidence indicators and $\theta_{x_i|u_i}$ be the set of parameter variables, then $C^{\oslash}$ should contain the following clauses:*

– *$\forall\ X_i \in V$, $C^{\oslash}$ contains the following two clauses (named indicator clauses):*

$$\lambda_{x_{i1}} \vee \lambda_{x_{i2}} \vee \cdots \lambda_{x_{in}} \tag{2}$$

$$\neg\lambda_{x_{ij}} \vee \neg\lambda_{x_{ik}}, j \neq k \tag{3}$$

– *$\forall\ \theta_{x_i|u_i}$ s.t $u_i = \{u_{i1}, u_{i2}, ..., u_{im}\}$, $C^{\oslash}$ contains the following clauses:*

$$\lambda_{x_i} \wedge \lambda_{u_{i1}} \wedge \ldots \wedge \lambda_{u_{im}} \rightarrow \theta_{x_i|u_i} \tag{4}$$

$$\theta_{x_i|u_i} \rightarrow \lambda_{x_i} \tag{5}$$

$$\theta_{x_i|u_i} \rightarrow \lambda_{u_{i1}}, \cdots, \theta_{x_i|u_i} \rightarrow \lambda_{u_{im}} \tag{6}$$

Clauses (2) and (3) state that indicator variables are exclusive, while clauses (4)-(6) encode network's structure. Once $f^{\oslash}$ is represented as $C^{\oslash}$, a compilation step is required to prepare for answering efficiently a large number of inference queries. This process depends on the uncertainty framework. Let $C_c^{\oslash}$ be the compilation result of $C^{\oslash}$. Let x be an instantiation of some variables $X \subseteq V$, then computing $\propto (x)$ using $C_c^{\oslash}$ is ensured as follows:

1. Conditioning $C_c^{\oslash}$ on x by setting each λ_{x_i} to $\bot$ if $\exists\ x_j \in x$ s.t. x_j and x_i disagree on values (i.e., $x_i \nsim x$), and to $\top$ if $x_i \sim x$.
2. Decoding $C_c^{\oslash}$ to have a valued expression, denoted by $\oslash$-circuit (Definition 2),
3. Computing $\propto (x)$ using $\oslash$-circuit.

Definition 2. *A $\oslash$-circuit is a DAG with internal nodes are labeled with $\oplus/\otimes$ and leaves are labeled with propositional variables.*

The first step allows us to exclude terms incompatible with x. The second step decodes the compiled base $\oslash$-circuit depending on the framework. The third step ensures an efficient computation of $\propto (x)$ by applying some query or transformation supported by some target compilation language. In the probabilistic case, inference problems have been effectively translated into *model counting* problems. According to the knowledge map of [7], the appropriate language is the sd-DNNF, which is less succinct than DNNF [7]. In the possibilistic case, inference corresponds to *max-variable elimination* (forgetting using the max), hence the language should support both max-variable elimination and conditioning. The honored language is Π-DNNF [1].

3.2 Which is the Most Compact Method?

In the previous subsection, we have only focused on network's structure and variable's domains and we have not explored parameters values, i.e., the so-called *local structure* which refers to a structure that can be inferred from the specific values of network parameters. Encoding local structure into logic has been under investigation in both probabilistic [3] and possibilistic settings [1]. We are in particular interested in parameters 0, 1 and equal parameters within CPTs. Incorporating local structure into $C^{\oslash}$ differs depending on the framework. In what follows, we will study in depth each case for both frameworks and reveal the differences between the two settings.

- *Parameters equal to 0*: Consider the parameter $\theta_{b_1|a_2} = 0$ which generates the three clauses of Definition 1. Given that this parameter is known to be 0, all terms that contain this parameter must vanish using either $\prod$ or min. Therefore, we can drop it from the encoding and replace its clauses by a shorter clause involving only indicator variables as follows: $\neg\lambda_{x_i} \vee \neg\lambda_{u_{i1}} \vee \cdots \vee \neg\lambda_{u_{im}}$.
- *Equal parameters*: Parameter equality is exploited to reduce at least the number of propositional variables. Due to the fact that no two parameters in the same CPT can ever appear in the same $f^{\oslash}$'s term, the same propositional variable can be used to represent multiple parameters within the same CPT. However, such simplification cannot be applied directly since an inconsistent family instantiations will be evoked at the level of clause (5) and clause (6). The idea is to drop these clauses from the encoding, which introduces additional models into the CNF, allowing multiple parameters from

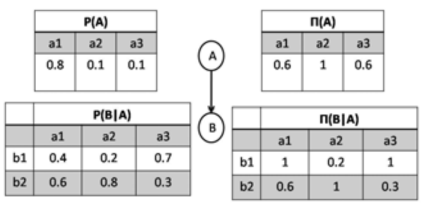
the same CPT in $f^{\oslash}$'s terms. These unintended models, having a cardinality higher than the cardinality of original models (i.e., $2N$: N evidence indicators and N network parameters), can be filtered by applying the *minimization* transformation that can be ensured in polytime by DNNF [4].

– *Parameters equal to 1*: Let $\theta_{x_i|u_i}$ be a parameter equal to 1. In the probabilistic case, we can omit this parameter and its associated clauses from C^B [5], while in the possibilistic case, this is not the case since the parameter 1, used to satisfy the normalization constraint, is not qualified as a particular value. Indeed, it is a fundamental parameter as it appears several times in each CPT. Hence, in the possibilistic case, parameters equal to 1 should be considered as a set of equal parameters within CPTs.

By exploiting local structure, the CNF encoding $C^{\oslash}$ and its corresponding circuit $\oslash$-circuit should be smaller, especially in the possibilistic case. In fact, the normalization constraint relative to the possibilistic network offers the opportunity to incorporate less propositional variables, encode less CNF clauses, and consequently construct more compact Π-circuits w.r.t. B-circuits, which proves the importance of the possibilistic setting versus the probabilistic setting.

Proposition 1. *Let Nb_{poss} and Nb_{proba} be the number of variables/clauses in the possibilistic and probabilistic cases, respectively. Then $Nb_{poss} \prec Nb_{proba}$.*

Example 1. *Let us consider the bayesian and the possibilistic networks of Fig. 1.*



P(A)		
a1	a2	a3
0.8	0.1	0.1

	P(B|A)		
	a1	a2	a3
b1	0.4	0.2	0.7
b2	0.6	0.8	0.3

Π(A)		
a1	a2	a3
0.6	1	0.6

	Π(B|A)		
	a1	a2	a3
b1	1	0.2	1
b2	0.6	1	0.3

Fig. 1. A bayesian and a possibilistic network

We stress that the number of propositional variables and clauses in C^{Π} is less than than those of C^B. In fact, there are 13 propositional variables and 28 clauses in C^B, while in C^{Π} there are only 11 variables and 22 clauses. The compilation of C^B and C^{Π} give us the following B-circuit and Π-circuit represented by figures 2 and 3, respectively s.t. θ_1 encodes the probability degree 0.1 and θ_1 (resp. θ_2) encode the possibility degree 0.6 (resp. 1). It is prominent that the number of nodes/edges of the Π-circuit is less than those of the B-circuit. Indeed, the number of nodes/edges in the probabilistic (resp. possibilistic) case is equal to 46/66 (resp. 41/60). These numbers include negated propositional variables. After evacuation of such parameters, the resulting compiled bases, which are represented by figures 2 and 3, contain 30/34 (resp. 25/31) nodes/edges in the probabilistic (resp. possibilistic) framework.

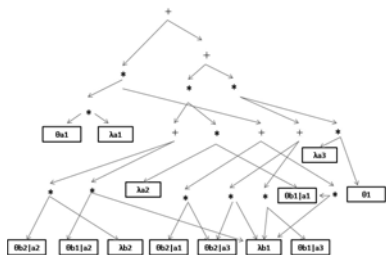


Fig. 2. The B-circuit

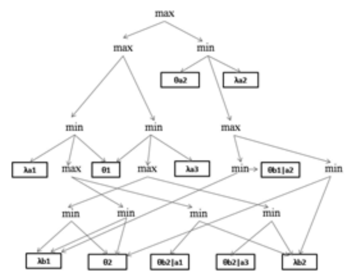


Fig. 3. The Π-circuit

4 Experimental Study

In this section, we will compare the probabilistic DNNF and the possibilistic Π-DNNF approaches. The objective behind this study is to highlight the extent to which possibility theory can reduce sizes of compiled bases. Our experimentation is performed on random bayesian and possibilistic networks generated as follows:

- *Graphical component*: DAGs are generated randomly, by just varying two parameters: the number of nodes and the maximum number of parents per node.
- *Numerical component*: Once the DAG structure is fixed for both bayesian and possibilistic networks, we generate random conditional probability and possibility distributions of each node in the context of its parents, with taking into consideration equal parameters and parameters equal to 0/1. These parameters will be stated as $\%EP$ the percent of equal parameters within the same CPT and $\%ExP$ the percent of extreme parameters within the same CPT. These two values give an idea on the amount of local structure within CPTs.

For each experimentation, we set $\%ExP$ to three values which are 10%, 50% and 90%. For each instantiation of $\%ExP$, we set $\%EP$ to 10%, 30%, 50% and 70%. For each instantiation of these parameters, we generate 100 random bayesian and possibilistic networks having a number of nodes equal to 50 and a maximum number of parents per node equal to 4. For each method, we will compare sizes of CNF encodings in terms of both number of variables and clauses and sizes of compiled bases in terms of both number of nodes and edges. Note that we have used the c2d compiler [6] developed by Darwiche. Each pair of the following figures show the behavior of DNNF and Π-DNNF for each instantiation of $\%EP$ and $\%ExP$. From figures 4, 6 and 8, we can deduce that the number of variables and clauses fall down in both DNNF and Π-DNNF for each instantiation of $\%EP$. Furthermore, it is prominent that Π-DNNF uses less variables and clauses comparing to DNNF. This is due to the normalization constraint offered by the possibilistic setting. Hence, the higher the value of $\%EP$ and $\%ExP$, the better the quality of results (i.e., the lower number of CNF variables and CNF clauses). By just taking a careful look at the scale of figures 4, 6 and 8

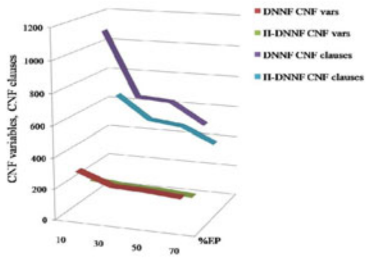


Fig. 4. Variables and clauses for $ExP = 10\%$

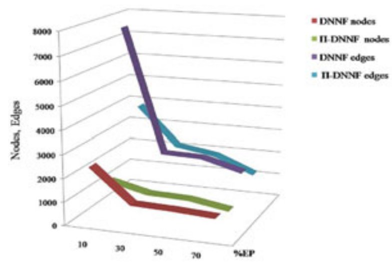


Fig. 5. Nodes and edges for $ExP = 10\%$

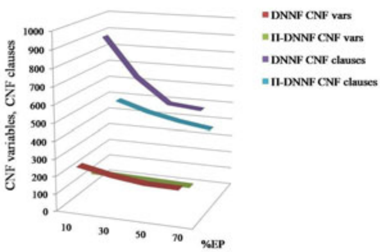


Fig. 6. Variables and clauses for $ExP = 50\%$

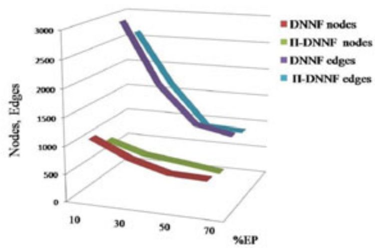


Fig. 7. Nodes and edges for $ExP = 50\%$

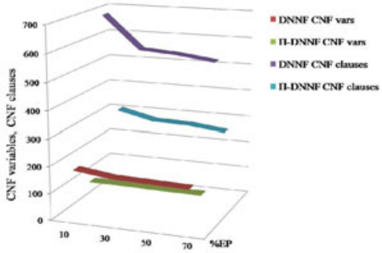


Fig. 8. Variables and clauses for $ExP = 90\%$

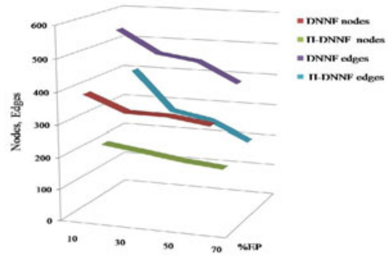


Fig. 9. Nodes and edges for $ExP = 90\%$

which decreases from 1200 to 700 through 1000, we can confirm this key result. Regarding compiled bases parameters, we can notice from figures 5,7 and 9 that Π-DNNF is characterized by a lower number of nodes and edges comparing to those of DNNF. Indeed, the higher the value of $\%EP$, the lower number of nodes and edges for both methods, especially for Π-DNNF. We should also pinpoint that compiled bases parameters and CNF parameters follow the same behavior since by increasing $\%EP$ or $\%ExP$, compiled bases parameters fall down, which is also the case for CNF parameters. Hence, we can conclude that Π-DNNF performs better than DNNF in terms of both CNF parameters and compiled bases parameters, which allows us to make up some extra space.

5 Conclusion

This paper proposed a generic compilation-based inference approach handling both probabilistic and possibilistic settings. We focused on theoretical in common points between the two approaches and unveil the differences between them. Furthermore, we studied the so-called local structure in both approaches. We theoretically show that Π-DNNF is more compact than DNNF, in terms of both number of CNF variables/clauses and nodes/edges of compiled bases, which proves the importance of the possibilistic setting versus the probabilistic setting. These results were confirmed by experimental results.

References

1. Ayachi, R., Ben Amor, N., Benferhat, S., Haenni, R.: Compiling possibilistic networks: Alternative approaches to possibilistic inference. In: Proceedings of 26th Conference on UAI, pp. 40–47. AUAI Press (2010)
2. Cadoli, M., Donini, F.M.: A survey on knowledge compilation. AI Communications–The EJAI (10), 137–150 (1998)
3. Chavira, M., Darwiche, A.: Compiling bayesian networks with local structure. In: Proceedings of the 19th IJCAI, pp. 1306–1312 (2005)
4. Darwiche, A.: Decomposable negation normal form. Journal of the ACM 48(4), 608–647 (2001)
5. Darwiche, A.: A logical approach to factoring belief networks. In: Proceedings of KR, pp. 409–420 (2002)
6. Darwiche, A.: New advances in compiling CNF to decomposable negational normal form. In: Proceedings of ECAI, pp. 328–332 (2004)
7. Darwiche, A., Marquis, P.: A knowledge compilation map. Journal of Artificial Intelligence Research 17, 229–264 (2002)
8. Dubois, D., Prade, H.: Possibility theory:An approach to computerized, Processing of uncertainty. Plenium Press, New York (1988)
9. Dubois, D., Prade, H.: Possibility theory. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and Systems Science, pp. 6927–6939. Springer, Heidelberg (2009)
10. Fargier, H., Marquis, P.: On valued negation normal form formulas. In: Proceedings of the 20th IJCAI, pp. 360–365 (2007)
11. Fonck, P.: Propagating uncertainty in a directed acyclic graph. In: Valverde, L., Bouchon-Meunier, B., Yager, R.R. (eds.) IPMU 1992. LNCS, vol. 682, pp. 17–20. Springer, Heidelberg (1993)