Extraction des politiques pour la programmation par ensembles réponses quantifiée

Martín Diéguez[1], Igor Stéphan[1]

[1] Université d'Angers, LERIA

martin.dieguezlodeiro | igor.stephan@univ-angers.fr

Résumé

La programmation par ensembles réponses quantifiée (Quantified Answer Set Programming, QASP) étend la programmation par ensembles réponses (Answer Set Programming, ASP) en autorisant la quantification des variables propositionnelles, à l'instar des QBF (Quantified Boolean Formulas) vis-à-vis de la logique propositionnelle. Dans cet article, nous interprétons les modèles de formules QASP en termes de politiques qui représentent des stratégies qui déterminent comme les variables existentiellement quantifiées doivent être assignées selon les conditions fixées par les variables universellement quantifiées. La contribution principale est un algorithme, qui extrait les politiques selon la sémantique QASP, inspiré par la sémantique de la logique de l'équilibre pour des théories ASP générales.

Mots-clés

QASP, quantification, ASP.

Abstract

Quantified Answer Set Programming (QASP) extends Answer Set Programming (ASP) by allowing quantification over propositional variables, similar to Quantified Boolean Formulas (QBF). In this paper, we interpret models of QASP formulas in terms of policies*, which represent decision-making strategies that determine how existentially quantified variables should be assigned, given the conditions set by universally quantified variables. As a main contribution, we present an algorithm for policy extraction under QASP semantics, inspired by the Equilibrium Logic semantics for general ASP theories.*

Keywords

QASP, quantification, ASP.

Ce que vous lirez en anglais

Ayant ses racines dans la programmation logique, la programmation par ensembles réponses (*Answer Set Programming* [5], ASP) est un formalisme pour le raisonnement non monotone (*nonmonotonic reasoning*, NMR) et la représentation de connaissances (*Knowledge Representation*, KR) qui résoult des problèmes complexes impliquant de la recherche combinatoire, de l'optimisation et du raisonnement dans l'incertain. En ASP, un problème est encodé en un ensemble de règles et les modèles (ensembles réponses) représentent les solutions. En marge de la traditionnelle sémantique des modèles stables, il existe des interprétations alternatives des programmes logiques selon la sémantique des ensembles réponses [7] incluant celle de la logique de l'équilibre (EL) de Pearce [8] qui utilise une base monotone et une condition de minimalité pour induire la non monotonie. Pour des théories propositionnelles, le problème de satisfaction en ASP se situe au second niveau de la hiérarchie polynomiale (Σ_2^p) [9]. Les approches telles que la sémantique stable-instable [4] étendent ASP pour résoudre les problèmes au-delà de cette classe de complexité en intégrant des programmes logiques comme des oracles. Une autre approche est le système ASP(Q) [1] [1], qui étend la syntaxe ASP avec des quantifications existentielles et universelles portant sur les modèles stables du programme plutôt que sur les atomes.

Dans le cas classique, les formules booléennes quantifiées (*Quantified Boolean Formulas* [3, Chapter 31], QBFs) étendent la logique propositionnelle avec des quantifications du second ordre autorisant l'expression de problèmes au-delà de la classe de complexité Σ_2^p, étant donné que le problème de satisfiabilité des QBF se situe dans PSPACE-complet [11]. La programmation par ensembles réponses quantifiée (QASP) [10, 6], similairement étend ASP en incorporant des quantificateurs propositionnels de manière analogue à QBF. Comme en QBF, les quantifications existentielles et universelles s'appliquent sur la valeur de vérité de l'atome propositionnel quantifié. La différence clé réside dans la sémantique : en QASP, la valeur de vérité assignée à l'atome doit être consistante avec les modèles stables du programme. Il existe deux sémantiques principales pour les programmes QASP. Dans la sémantique proposée par Fandinno et al. [6], un atome p est forcé a être vrai dans le modèle stable en ajoutant la formule $\neg\neg p$ au contexte. En revanche, dans la sémantique définie dans Stephan [10], p est rendu vrai simplement en ajoutant p au contexte. Les deux sémantiques sont identiques lorsqu'il s'agit de déterminer si un atome est faux dans un modèle stable. En termes d'implantation, la sémantique de [6] est réalisée en un outil appelé `qasp2qbf` [2], qui a été utilisé pour résoudre des

1. https://www.mat.unical.it/ricca/downloads/qasp-0.1.2.jar
2. https://github.com/potassco/qasp2qbf.

problèmes de planification sous incertitude.

En QASP, les solutions à un programme logique quantifié peuvent être interprétées comme des stratégies dans un jeu à deux joueurs avec information complète. Le premier joueur tente de construire les modèles stables du programme ASP, et le programme est considéré comme une instance positive du problème QASP si et seulement si une stratégie gagnante pour le premier joueur. L'outil `qasp2qbf` réduit un programme QASP à une QBF, dont la satisfiabilité est décidée par un solveur QBF. Si la QBF est satisfiable, le solveur fournit un assignement pour le bloc initial de quantificateurs existentiels. Cela signifie que si le programme QASP commence avec un quantificateur universel, le solveur ne peut calculer que le caractère satisfiable ou non, sans produire une stratégie. En revanche, l'approche présentée dans [10] remplace les quantificateurs existentiels avec des fonctions de Skolem explicitant les choix du joueur existentiel. Cette méthode est donc en mesure de calculer toutes les stratégies qui satisfont le programmme QASP.

Dans le présent article, nous explorons les sémantiques de [6] et de [10] sous la perspective de la logique de l'équilibre. Nous introduisons la restriction suivante : nous ne considérons pas un langage logique spécifique mais, à la place, nous traduisons chaque règle d'un programme logique $(H \leftarrow B)$ en une implication $(B \rightarrow H)$ en logique propositionnelle. La négation par défaut (`not`) est remplacée par la négation ($\neg$) de la logique propositionnelle, et les formules propositionnelles peuvent être imbriquées de manière arbitraire.

La première contribution de cet article démontre que les deux sémantiques ne sont pas équivalentes. La deuxième contribution est l'extension de la notion de politique du cas classique à la logique propositionnelle de Gödel quantifiée QG_3 [2], qui fournit une base monotone pour sélectionner les politiques minimales. La troisième contribution est un algorithme qui prend une QBF en entrée et calcule l'ensemble des politiques selon la sémantique de [6]. Cet algorithme combine les politiques QBF et QG_3 durant l'exécution.

Références

[1] G. Amendola, F. Ricca, and M. Truszczynski. Beyond NP : Quantifying over answer sets. *Theory and Practice of Logic Programming*, 19(5-6) :705–721, 2019.

[2] M. Baaz, A. Ciabattoni, and R. Zach. Quantified propositional Gödel logics. In *Proceedings of the 7th International Conference on Logic for Programming and Automated Reasoning (LPAR)*, volume 1955 of *LNCS*, pages 240–256. Springer, 2000.

[3] A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability*, volume 336 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2021.

[4] B. Bogaerts, T. Janhunen, and S. Tasharrofi. Stable-unstable semantics : Beyond NP with normal logic programs. *Theory and Practice of Logic Programming*, 16(5-6) :570–586, 2016.

[5] G. Brewka, T. Eiter, and M. Truszczyński. Answer set programming at a glance. *Communications of the ACM*, 54(12) :92–103, 2011.

[6] J. Fandinno, F. Laferriere, J. Romero, T. Schaub, and T. Son. Planning with incomplete information in quantified answer set programming. *Theory and Practice of Logic Programming*, 21(5) :663–679, 2021.

[7] V. Lifschitz. Thirteen definitions of a stable model. In *Fields of Logic and Computation : Essays Dedicated to Yuri Gurevich on the Occasion of His 70th Birthday*, pages 488–503. Springer, 2010.

[8] D. Pearce. Equilibrium logic. *Annals of Mathematics and Artificial Intelligence*, 47(1-2) :3–41, 2006.

[9] D. Pearce, H. Tompits, and S. Woltran. Encodings for equilibrium logic and logic programs with nested expressions. pages 306–320.

[10] I. Stéphan. QASP ou la programmation par ensembles réponses quantifiée. In *Actes du dix-neuvième congrès national sur la Reconnaissance de Formes et l'Intelligence Artificielle (RFIA)*, 2014. In French.

[11] L.J. Stockmeyer and A.R. Meyer. Word problems requiring exponential time. In *Proceedings of the 5th annual ACM symposium on Theory of computing (STOC)*, pages 1–9, 1973.