MDV : arithmétique et procédures de décision

Plan

Plan

Plan

Plan

Plan

Motivations : omniprésence de l'arithmétique

- Types : entiers, énumérations, vecteurs
- Opérations : e.g., modifier, tester la valeur d'un compteur

Comme pour la logique, besoin de formaliser l'arithmétique.

Arithmétique : syntaxe et sémantique

Syntaxe

- constante 0, variables, fonction unaire S, fonctions binaires $+$, $*$, prédicat $=$
- termes : application des symboles de fonctions aux constantes
- formule élémentaire : application de $=$ aux termes
- formule : formule élémentaire, connecteurs logiques, quantificateurs
- abréviations : e.g., $2 \equiv S(S(0))$, $y < x \equiv \exists z, (y + z = x) \ldots$

Sémantique

- domaine : entiers naturels
- interprétations usuelles pour tous les symboles
- modèle : valuation des variables libres qui rendent la formule *vraie*
  Ex : $y = 1$ pour $\exists x.(y < x)$, car il existe $x = 2$ tel que $1 < 2$

Axiomes de Peano

Règles de la logique du premier ordre, plus :

1. $\forall x.\ \neg(0 = S(x))$
2. $\forall x \forall y.\ \neg(x = y) \Rightarrow \neg(S(x) = S(y))$
3. $\forall x.\ x + 0 = x$
4. $\forall x \forall y.\ S(x + y) = x + S(y)$
5. $\forall x.\ x * 0 = 0$
6. $\forall x \forall y.\ x * S(y) = x * y + x$
7. pour toute formule P avec x libre :
   $(P(0) \land \forall x.\ (P(x) \Rightarrow P(S(x)))) \Rightarrow \forall x.\ P(x)$

Preuves dans l'arithmétique de Peano

Preuve de F : séquence de formules $F_1, \dots, F_n$ avec $F_n = F$ et pour tout $i = 1, \dots, n$: F_i obtenu de $\{F_j \mid 1 \leqslant j < i\}$ en utilisant un axiome.

Exemple : preuve de $\forall x.\ x + 0 = 0 + x$.

- Axiome 3 réduit l'énoncé à $\forall x.\ x = 0 + x$
- Axiome 7 réduit l'énoncé à $\forall x.(x = 0 + x) \Rightarrow (S(x) = (0 + S(x)))$
- Par l'Axiome 4 : $S(0 + x) = 0 + S(x)$, donc l'énoncé devient $\forall x.(x = 0 + x) \Rightarrow (S(x) = S(0 + x))$
- Logique du premier ordre pour conclure.

Vrai = Prouvable en Peano ?

- Cohérence : F (close) prouvable en Peano implique F sémantiquement valide

- Complétude : F (close) valide implique prouvable en Peano ?
  (Non... cf. premier théorème d'incomplétude de Gödel, 1931)

Théorème d'incomplétude de Gödel

Théorème

Si T est une théorie non contradictoire, contenant les axiomes de l'arithmétique, alors T est incomplète (il existe des formules sémantiquement valides mais non prouvables).

Conséquences :

- impossible d'axiomatiser complètement l'arithmétique élémentaire !
- le raisonnement purement formel est plus faible que le raisonnement mathématique habituel.

« Preuve » du théorème d'incomplétude de Gödel

Idée : construire dans l'arithmétique un énoncé F : « F n'est pas prouvable », en codant les formules avec des entiers

Par l'absurde : supposer la complétude de Peano.

Donc : F est prouvable ssi F est valide ssi F n'est pas prouvable... contradiction.

Indécidabilité de l'arithmétique

Problème de satisfiabilité : une formule de Peano admet-elle un modèle ?

Le problème de satisfiabilité est indécidable (ce qui implique aussi l'incomplétude).

Preuve de l'indécidabilité : par réduction d'un problème connu indécidable à la satisfiabilité.

L'accessibilité dans les machines à compteurs

$$
\begin{tikzcd}[column sep=large]
{} \arrow[r, "\begin{array}{c} x := 0 \\ y := 0 \end{array}"] & pc = 1 \arrow[r, bend right, "y \leqslant 0 \to x := x+1, y := y+1"'] & pc = 2 \arrow[l, bend right, "x > 0 \to y := y-1"'] \arrow[r, "x = 0"] & pc = 3
\end{tikzcd}
$$

- Relation de transition : $\rho(pc, x, y, pc', x', y')$ entre valeurs avant/après.
- Exécution : séquence de triplets suivant la relation de transition.
- Accessibilité : existence d'une exécution menant à un état donné : *indécidable*.

Réduction accessibilité → satisfiabilité

Accessibilité de $pc = k$ ssi il existe $n \in \mathbb{N}$ et trois séquences PC, X, Y :
$PC(1) = 1 \wedge PC(n) = k \wedge \forall i.(0 \leqslant i < n \Rightarrow \rho(PC(i), X(i), Y(i), PC(i+1), X(i+1), Y(i+1)))$

Codage des trois séquences par un nombre naturel :
$N = 2^{PC(1)} \cdot 3^{X(1)} \cdot 5^{Y(1)} \cdots p_{3n+1}^{PC(n)} \cdot p_{3n+2}^{X(n)} \cdot p_{3n+3}^{Y(n)}$

Décodage : $e(N, 3i+1) = PC(i)$, $e(N, 3i+2) = X(i)$, $e(N, 3i+3) = Y(i)$
ρ, codage, décodages : fonctions *exprimables en arithmétique de Peano*.

Plan

Une arithmétique décidable : Presburger

- Même syntaxe et sémantique que Peano, mais pas de multiplication (la multiplication par une constante est permise, c'est une suite finie d'additions)
- Exemple : $\forall x. \exists y. y \geqslant 2 * x$
- Décider la satisfiabilité : par élimination des quantificateurs : P' équivalente à P, ne contient plus que des constantes : évaluation par calcul arithmétique.

Procédure de décision

Transformer chaque $\forall x.F(x)$ en $\neg(\exists x.\neg F(x))$ puis, tant qu'il existe des quantificateurs, itérer

- mise en forme normale disjonctive
- élimination des négations élémentaires, i.e., $\neg(x < y)$ devient $y < x + 1$
- distribution de $\exists$ sur $\vee$
- éliminer $\exists$ de $\exists x.F(x)$, ou F est une conjonction d'(in)équations affines et de relations $div(c, t)$ ou $\neg div(c, t)$ (c constante, t terme affine).
  - Exemple : si $F(x, y) = (x = 2y)$, alors $\exists y.F(x, y) \equiv div(2, x)$.
  - utilisation des propriétés du *p.p.c.m.*

Arithmétique : résumé

- L'arithmétique est indécidable et incomplète pour tout système raisonnable d'axiomes : en général, un humain doit guider la preuve.
- L'arithmétique de Presburger est décidable (et complète pour le sous-ensemble correspondant des axiomes de Peano). Complexité triple exponentielle déterministe ($2^{2^{2^{cn}}}$), mais simple sans quantificateurs.

Biblio :

- Lassaigne & Rougemont, *Logique et Fondements de l'informatique*, chapitre 8
- David, Nour & Raffalli, *Introduction à la logique*, chapitre 3
- `http ://www.cs.umd.edu/projects/omega/` (projet Omega)

Presburger + symboles de fonctions (PF)

Pour modéliser les vecteurs, les séquences : fonctions de $\mathbb{N}^n$ dans $\mathbb{N}$.

Exemple : $\forall y.(f(y) > y \land \exists z.(z + 1 < 2f(x + f(y + 1))))$

Modèle : donnée d'une valeur pour chaque variable libre, et d'une interprétation pour chaque fonction, qui satisfont la formule

- Satisfiabilité indécidable [Halpern, 1991],
- mais décidable dans le fragment sans quantificateurs [Shostak, 1979].

Décidabilité du fragment de PF sans quantificateurs

Propriété : les fonctions envoient des égaux sur des égaux

- remplacer chaque $f(t)$ par une variable entière fraîche f_t
- utiliser la Propriété pour contraindre les variables fraîches

Exemple :

$$u = v \land f(u) \neq f(v) \text{ satisfiable}$$
$$\Longleftrightarrow$$
$$u = v \land f_u \neq f_v \land (u = v \Rightarrow f_u = f_v) \text{ satisfiable}$$
$$\Longleftrightarrow$$
$$false \text{ satisfiable}$$

Résumé et suite

- Arithmétique (de Peano, de Presburger, avec fonctions) décidabilité et indécidabilité(s)
- il existe d'autres théories décidables, utiles pour les programmes
- on peut combiner leurs procédures de décision.

Plan

Théories des réels

Théorème

La théorie du 1er ordre des réels avec addition et multiplication *est décidable [Tarski].*

- Meilleur algorithme connu : [Collins] $L^3(md)^{2^{O(n)}}$, où :
  L = taille coeffs., m = nb. contraintes, d = degré max, n = nb. variables.
- Fonctionne par élimination de quantificateurs, exemple :
  $a > 0 \land \exists x.(ax^2 + bx + c = 0)$ simplifiée en $a > 0 \land b^2 - 4ac \geqslant 0$
- Complexité trop élevée en pratique (sauf fragment sans multiplication)

Théorie monadique du second ordre avec successeur

- Permet de quantifier sur des ensembles finis.
- Exemple : x est pair :

$$\exists Q.(Q(0) \wedge Q(x) \wedge \forall q.(0 \leqslant q < x) \Rightarrow (Q(q) \iff \neg Q(S(q))))$$

- Traduction vers le formalisme des automates finis
- Complexité non élémentaire : $2^{2^{2^{\cdots}}}$ de hauteur proportionnelle à la longueur de la formule... mais bons résultats en pratique !
- Projet Mona : `http ://www.brics.dk/mona/`

Théorie des fonctions non interprétées avec égalité

- Fonctions totales aux domaines et co-domaines non interprétés.
- Sans quantificateurs, seul axiome non-logique : $x = y \Rightarrow f(x) = f(y)$
- Exemple : $x = f(f(x)) \Rightarrow x = f(f(f(f(x))$
- Théorie introduite pour la vérification de processseurs
- Décision [Nelson] : calcul de classes d'équivalences de termes.

Théorie des listes (à la LISP)

- Sans quantificateurs, axiomes non-logiques :
  $car(cons(x, y)) = x$
  $cdr(cons(x, y)) = y$
  $\neg atom(x) \Rightarrow cons(car(x), cdr(x)) = x$
  $\neg atom(cons(x, y))$
- Décision : par réécriture (généralisation : types abstraits algébriques)

Combinaison de procédures de décision

- La plupart des objectifs de vérification s'expriment en plusieurs théories.
- Exemple : $x \leqslant y \wedge y \leqslant x + car(cons(0, x)) \wedge P(h(x) - h(y)) \wedge \neg P(0)$
- Utilise les théories $\mathcal{R}$ de réels, $\mathcal{L}$ des listes, et $\mathcal{F}$ des fonctions.
- Décision : propagation d'égalités entre les procédures de $\mathcal{R}$, $\mathcal{L}$, $\mathcal{F}$ [Nelson, Oppen 1979]

Exemple de propagation d'égalités

$$x \leqslant y \wedge y \leqslant x + car(cons(0, x)) \wedge P(h(x) - h(y)) \wedge \neg P(0)$$

Séparation des formules par théorie par introduction de nouvelles variables :

$\mathcal{R}$	$\mathcal{F}$	$\mathcal{L}$
$x \leqslant y$ $y \leqslant x + g_1$ $g_2 = g_3 - g_4$ $g_5 = 0$	$P(g_2) = true$ $P(g_5) = false$ $g_3 = h(x)$ $g_4 = h(y)$	$g_1 = car(cons(g_5, x))$

Exemple de propagation d'égalités (2)

$\mathcal{L}$ détecte $g_1 = g_5$, propage cette égalité à $\mathcal{R}$ et $\mathcal{F}$:

$\mathcal{R}$	$\mathcal{F}$	$\mathcal{L}$
$x \leqslant y$	$P(g_2) = true$	$g_1 = car(cons(g_5, x))$
$y \leqslant x + g_1$	$P(g_5) = false$	$\mathbf{g_1 = g_5}$
$g_2 = g_3 - g_4$	$g_3 = h(x)$	
$g_5 = 0$	$g_4 = h(y)$	

Exemple de propagation d'égalités (3)

$\mathcal{R}$ utilise $g_1 = g_5$, détecte $x = y$, propage cette égalité à $\mathcal{F}$ et $\mathcal{L}$:

$\mathcal{R}$	$\mathcal{F}$	$\mathcal{L}$
$x \leqslant y$	$P(g_2) = true$	$g_1 = car(cons(g_5, x))$
$y \leqslant x + g_1$	$P(g_5) = false$	$\mathbf{g_1 = g_5}$
$g_2 = g_3 - g_4$	$g_3 = h(x)$	
$g_5 = 0$	$g_4 = h(y)$	
$\mathbf{x = y}$		

Exemple de propagation d'égalités (4)

$\mathcal{F}$ utilise $x = y$, détecte $g_3 = g_4$, propage cette égalité à $\mathcal{R}$ et $\mathcal{L}$:

$\mathcal{R}$	$\mathcal{F}$	$\mathcal{L}$
$x \leqslant y$	$P(g_2) = true$	$g_1 = car(cons(g_5, x))$
$y \leqslant x + g_1$	$P(g_5) = false$	$\mathbf{g_1 = g_5}$
$g_2 = g_3 - g_4$	$g_3 = h(x)$	
$g_5 = 0$	$g_4 = h(y)$	
$\mathbf{x = y}$	$\mathbf{g_3 = g_4}$	

Exemple de propagation d'égalités (5)

$\mathcal{R}$ utilise $g_3 = g_4$, détecte $g_2 = g_5$, propage cette égalité à $\mathcal{F}$ et $\mathcal{L}$:

$\mathcal{R}$	$\mathcal{F}$	$\mathcal{L}$
$x \leqslant y$	$P(g_2) = \mathit{true}$	$g_1 = \mathit{car}(\mathit{cons}(g_5, x))$
$y \leqslant x + g_1$	$P(g_5) = \mathit{false}$	$\mathbf{g_1 = g_5}$
$g_2 = g_3 - g_4$	$g_3 = h(x)$	
$g_5 = 0$	$g_4 = h(y)$	
$\mathbf{x = y}$	$\mathbf{g_3 = g_4}$	
$\mathbf{g_2 = g_5}$		

Exemple de propagation d'égalités (6)

$\mathcal{F}$ utilise $g_2 = g_5$, détecte une incohérence : la formule est insatisfiable !

$\mathcal{R}$	$\mathcal{F}$	$\mathcal{L}$
$x \leqslant y$	$P(g_2) = true$	$g_1 = car(cons(g_5, x))$
$y \leqslant x + g_1$	$P(g_5) = false$	$\mathbf{g_1 = g_5}$
$g_2 = g_3 - g_4$	$g_3 = h(x)$	
$g_5 = 0$	$g_4 = h(y)$	
$\mathbf{x = y}$	$\mathbf{g_3 = g_4}$	
$\mathbf{g_2 = g_5}$	$\mathbf{true = false}$	

Plan

Tactiques arithmétiques en Coq

- `ring` : résolution d'équations polynomiales sur un anneau ou un semi-anneau : égalités où les membres sont construits avec addition, multiplication, opposé, et variables prises sur un anneau (par exemple `Z` ou `nat`)
- `omega` : systèmes d'équations linéaires sur `Z` et `nat`
- `field` : idem `ring`, mais sur les corps (avec division)
- `fourier` : idem `omega`, mais sur les réels

Exercice

Toute somme de plus de 8 euros peut être payée en « pièces » de 3 et 5 euros :

$$\forall n \in \mathbb{N}, \exists i, j \in \mathbb{N}, (n + 8 = 5i + 3j)$$

Par induction :

- si $n = 0$ alors $i = 1, j = 1$
- supposons que $n + 8 = 5i + 3j$, le prouver pour $n + 1$. Raisonnement par cas :
  - si $j \geqslant 3$ alors $n + 9 = 5(i + 2) + 3(j - 3)$
  - si $i \geqslant 1$ alors $n + 9 = 5(i - 1) + 3(j + 2)$
  - autrement ($i = 0$ et $j \in \{0, 1, 2\}$) : cas inutiles.

Plan

Preuves par réflexion

- Idée : remplacer des étapes de preuve par des étapes de calcul
- Exemple : prouver des propriétés telles que

$$(4 * x + (8 * x + (15 * x + (16 * x + (23 * x + 42 * x))))) = 108 * x)$$

- Soit faire une preuve directe, en utilisant 5 fois la distributivité
- Soit faire la preuve pour tout couple (l, s) où l est une liste d'entiers telle que la somme de ses éléments vaille s

Preuves par réflexion : résumé

- Une « syntaxe » des formules : type `F`
- Une sémantique des formules : `sem : F -> Prop`
- Un prouveur qui calcule une valeur de vérité : `F -> bool`
- Un théorème de correction du prouveur : `prover_sound`
- Pour obtenir une preuve sur un cas particulier :
  - reconnaître dans le but la sémantique d'une formule et la remplacer par sa forme syntaxique
  - appliquer le théorème de correction
  - calculer

Conclusion

- Les procédures de décision sont indispensables pour automatiser les parties « faciles » des preuves.
- Il n'est pas toujours facile d'isoler les fragments décidables.
- Certains outils de preuve utilisent des procédures de décision « externes ». Quid de leur fiabilité ?
- La réflexion en Coq permet d'implanter ses propres procédures.