Domaine abstrait des signes et produit réduit

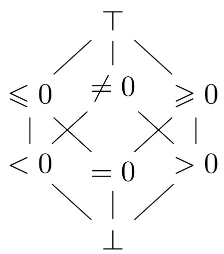
Préliminaires

L'analyseur devrait avoir déjà été récupéré et installé lors des TP, sinon se reporter à la section « Préliminaires » du premier TP. Récupérer de plus les fichiers nécessaires pour ce BE et les placer dans le dossier `src/domains` de l'analyseur :

```
> cd dossier/ou/on/a/mis/tiny
> cp archive_du_be/source/* src/domains/
```

Domaine des signes (15 points)

Compléter le fichier `signes.ml` pour obtenir une implémentation du domaine des signes, qui est une variation (plus précise) du domaine vu en cours. On donne ci-dessous le treillis sous-jacent au domaine ainsi que sa fonction de concrétisation. On constate que le treillis est un cube, i.e. le produit de trois treillis à deux valeurs.



$$
\begin{array}{lcl}
\gamma(\top) & = & \mathbb{Z} \\
\gamma(\neq 0) & = & \{n \in \mathbb{Z} \mid n \neq 0\} \\
\gamma(\leqslant 0) & = & \{n \in \mathbb{Z} \mid n \leqslant 0\} \\
\gamma(\geqslant 0) & = & \{n \in \mathbb{Z} \mid n \geqslant 0\} \\
\gamma(< 0) & = & \{n \in \mathbb{Z} \mid n < 0\} \\
\gamma(> 0) & = & \{n \in \mathbb{Z} \mid n > 0\} \\
\gamma(= 0) & = & \{\, 0 \,\} \\
\gamma(\bot) & = & \emptyset
\end{array}
$$

Pour compiler, ne pas oublier de modifier la première ligne de code du fichier `src/analyze.ml`. Il est bien sûr fortement conseillé de corriger tout warning qui apparaîtrait à la compilation et de tester le domaine implémenté, au moins sur les fichiers du dossier `examples`.

Précisions sur la notation : Chaque fonction à écrire ou à compléter est notée sur 2 points (sauf la dernière sur 3 points). Une fonction incorrecte sera notée 0 et toute fonction correcte se verra attribuer une note entre 0 et 2 (ou 3) suivant sa précision [1].

Produit réduit (5 points + 1 point bonus)

Attention : Cet exercice est beaucoup plus difficile que le précédent, *ne l'entamer qu'une fois toutes les fonctions du domaine des signes correctement implémentées*. Pour garantir cela, cet exercice ne sera pas corrigé si la note du précédent est inférieure à 10.

Produit cartésien de domaines abstraits

Si $(\mathcal{D}_1, \sqsubseteq_1)$ et $(\mathcal{D}_2, \sqsubseteq_2)$ sont deux treillis alors leur produit cartésien $\mathcal{D}_1 \times \mathcal{D}_2$ est un treillis muni de l'ordre : $(x_1, x_2) \sqsubseteq (y_1, y_2)$ si $x_1 \sqsubseteq_1 y_1$ et $x_2 \sqsubseteq_2 y_2$. Et si on forme des domaines abstraits en munissant ces treillis d'opérations abstraites (par exemple $+^\sharp_1$ et $+^\sharp_2$), le produit cartésien

1. 0 pour la fonction constante $\top$ et 2 (ou 3) pour la fonction optimale par exemple.

forme un domaine abstrait muni des opérations composante par composante (pour continuer l'exemple $+^\sharp$ sera défini par $(x_1^\sharp, x_2^\sharp) +^\sharp (y_1^\sharp, y_2^\sharp) = (x_1^\sharp +_1^\sharp y_1^\sharp, x_2^\sharp +_2^\sharp y_2^\sharp))$.

On fournit un foncteur OCAML `NonRelationalProduct.Make` réalisant ce produit cartésien. Faire le produit cartésien des domaines intervalles et parité fournis en remplaçant la première ligne de code du fichier `src/analyze.ml` par la suivante :

```
module Dom : Relational.Domain = NonRelational.MakeRelational
  (NonRelationalProduct.Make (Parity) (Intervals))
```

et tester ce nouveau domaine sur `examples/ex11.tiny`.

— L'analyseur signale une possible division par zéro, s'agit il d'une véritable erreur ou d'une fausse alarme ? Justifier la réponse.
— Quelle est la source du problème ?
— Comment pourrait on y remédier avec les informations données par les domaines parité et intervalles ?

Vous répondrez à ces question au début du fichier `produitPariteIntervalles.ml` en quelque lignes [2] de commentaires.

Réduction

Il arrive – en particulier après avoir effectué un produit cartésien de deux domaines – que plusieurs valeurs abstraites aient la même concrétisation. Il est alors intéressant de les remplacer par une représentation ≪ canonique ≫. Par exmple, si on effectue le produit cartésien du domaine des signes implémenté dans le premier exercice avec le domaine parité vu à la question précédente, les valeurs abstraites $(0, \mathit{impair})$, $(\bot, \mathit{pair})$, $(\geqslant 0, \bot)$ et $(\bot, \bot)$ ont toutes quatre pour concrétisation l'ensemble vide $\emptyset$. On voudrait donc les remplacer par $(\bot, \bot)$.

Plus formellement, une réduction d'un domaine abstrait $\mathcal{D}^\sharp$ est une fonction ρ du domaine dans lui même vérifiant les conditions suivantes :

— $\forall x^\sharp \in \mathcal{D}^\sharp,\ \gamma(\rho(x^\sharp)) = \gamma(x^\sharp)$ (la réduction est correcte...) ;
— $\forall x^\sharp \in \mathcal{D}^\sharp,\ \rho(x^\sharp) \sqsubseteq^\sharp x^\sharp$ (...et c'est bien une réduction).

où γ est la fonction de concrétisation du domaine $\mathcal{D}^\sharp$ et $\sqsubseteq^\sharp$ l'ordre sur ce domaine.

Définir *mathématiquement* une réduction ρ du produit cartésien des domaines parité et intervalles vu lors des questions précédentes [3] puis implémenter cette réduction en complétant la fonction `rho` du fichier `produitPariteIntervalles.ml`. L'analyse signale t-elle toujours un risque de division par zéro sur le fichier `examples/ex11.tiny` ?

Rendu

Pour rendre votre travail, exécuter le script fourni et suivre ses instructions pour l'envoi de l'archive créée par courrier électronique :

```
> src/domains/archive_rendu.sh
```

N.B. : L'archive doit être envoyée avant 13h00, sous peine de perte de points.

2. Nul besoin d'écrire un roman.
3. Encore une fois, on répondra sous forme de commentaire. D'autres part, cette réduction doit être ≪ intéressante ≫ (la fonction identité est une réponse correcte mais ne rapportera pas beaucoup de points).