CentraleSupélec

TP Frama-C / WP & MetAcsl – le 03 avril 2023
Nikolaï Kosmatov

L'objectif de ce TP est la découverte des techniques de preuves à base de métapropriétés, d'axiomatiques, de fonctions logiques, de lemmes et de script. Il vise à compléter la spécification et à prouver le module memb de Contiki, un OS pour l'Internet des Objets. Un compte rendu de l'exercice 2 sera rendu après la séance.

Exercice 1. Découverte de MetAcsl

Nous allons spécifier à l'aide de métapropriétés (HILAREs) les propriétés dans le fichier ex1.c et essayer de les prouver. La commande à lancer et les propriétés à spécifier sont indiquées dans le fichier. Observez le résultat de transformation de propriétés réalisée par MetAcsl et les résultats de preuve pour chaque exemple.

Exercice 2. Preuve de memb

Nous allons compléter la spécification et prouver le module memb contenu dans le fichier memb.c. On commencera par parcourir le fichier memb.c pour avoir une idée des fonctions. Des commandes utiles sont indiquées au début du fichier.

Un fichier supplémentaire OccArray.acsl contiendra une définition axiomatique de la fonction logique occ_a et des lemmes utiles pour la preuve.

La structure memb décrit un tableau pré-alloué de blocs (éléments) avec un nombre donné de blocs (dans le champ num) dont chacun a la même taille (champ size). Ils peuvent être attribués ou libérés sur demande. Le tableau des blocs (éléments) est représenté par le pointeur du champ mem. Le statut alloué/libre des blocs est représenté par le tableau du champ count (valeur 0 indique le statut libre, sinon attribué).

1. Ecrire la définition du prédicat valid_memb dans le fichier memb.c qui exprime la validité d'une structure memb pointé par son argument. Donner et expliquer la définition proposée.

2. Ecrire la définition de la fonction logique _memb_ptr dans le fichier memb.c qui retourne le pointeur vers le bloc (élément) de l'indice donné dans la structure. Donner et expliquer la définition proposée.

3. Compléter la définition de l'axiomatique OccArray dans le fichier OccArray.acsl pour le cas manquant afin de définir le compteur d'occurrence d'un nombre donné dans un tableau de nombres. occ_a(elt, tab, from, to) retourne le nombre d'occurrence de elt dans tab entre indice from (inclus) et to (exclu). Donner et expliquer la définition proposée.

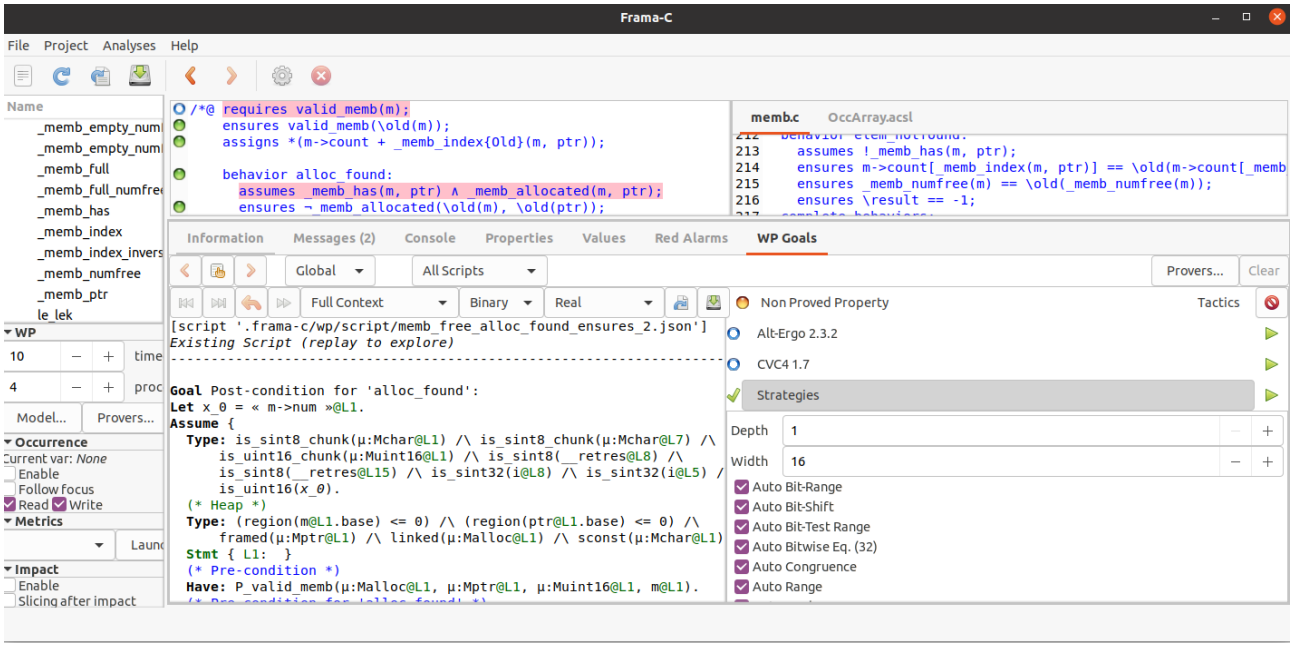
4. Compléter le contrat de la fonction memb_free (le contrat de boucle donnée devrait être suffisant). Donner et expliquer la définition proposée. Essayer de prouver la fonction.

5. Ecrire le contrat de la fonction memb_numfree ainsi que le contrat de boucle. Expliquer les contrats proposés. Essayer de prouver la fonction. Indication. On pourra utiliser la fonction

logique _memb_numfree(m) pour décrire la valeur retournée par la fonction. On pourra utiliser occ_a(0, m->count, 0, i) pour décrire la valeur de num_free courante dans l'invariant de boucle.

6. Spécifier une métapropriété indiquant que les cases du tableau des statuts ne peuvent être modifiées dans les fonctions **memb_inmemb,memb_numfree**. Essayer de la prouver. Notez que puisque le pointeur vers la structure memb m est un argument de ces fonctions (et non pas une variable globale), dans une métapropriété il faut mettre \formal(m) au lieu de m pour éviter des erreurs de syntaxe.

7. (sans réponse écrite) On pourra observer et refaire le script de preuve pré-défini pour une des postconditions de la fonction memb_free. Afficher la liste des tous les goals. Faites un double clic sur le nom de la propriété prouvée avec un script pour entrer dans l'interface de création de script. Supprimer la preuve existante (bouton rouge à droite) puis cliquer sur « Strategies » pour tenter la création d'un nouveau script. Si la preuve est réussie, vous pouvez enregistrer le script (il sera enregistré par défaut dans le dossier .frama-c/wp/script). Un script enregistré sera rejoué lors d'une nouvelle session de Frama-C si « script » fait partie des arguments de l'option : **-wp-prover=script**,...



8. (sans réponse écrite, les scripts Coq fournis ont été testés avec Frama-C 23) On pourra observer (modifier) la preuve d'un lemme écrit en ACSL et prouvée dans Coq. Afficher la liste des tous les goals. Dans la colonne Coq, après un clic droit sur la pastille du lemme « everywhere_means_all_occ_a », choisir le menu « Edit proof ». On voit s'ouvrir coq-ide. On peut modifier et enregistrer la preuve. Elle sera enregistrée dans le fichier wp.script (avec un numéro en plus si le fichier existe déjà). Les preuves Coq sont rejouées lors d'une nouvelle session de Frama-C si on donne e.g. l'option : -wp-prover=script,alt-ergo,cvc4,**native:coq -wp-coq-script wp.script**
Note : deux versions de ce script sont fournies, pour Coq 8.12.0 et Coq 8.11.10, appelées wp_coq.8.12.0.script et wp_coq.8.11.10.script. Pour connaître votre version, taper coqc -v.
9. (facultatif) On pourra essayer de prouver dans Coq quelques-uns des lemmes non prouvés et soumettre la preuve enregistrée dans le fichier wp.script avec votre rapport.