le cnam
Ile-de-France

le cnam
Ile-de-France

Projet Web (NFA021)

Avril 2014

Cnam de Saint-Denis

Réalisation d'une interface d'exécution et de comparaison d'outils de déduction automatique

1 Contexte et objectifs

Ce projet s'inscrit dans un projet de recherche et consistera à réaliser une interface web permettant de comparer les performances de deux outils expérimentaux. Ces deux outils en question sont des outils de déduction automatique, c'est-à-dire des outils qui cherchent à démontrer automatiquement des problèmes. Pour réaliser ce projet, aucune connaissance sur la théorie de ces outils ou sur la forme des problèmes que prennent ces outils en entrée ne sera nécessaire. Il suffit juste de comprendre que ces outils prennent un problème en entrée et répondent « oui » s'ils ont réussi à trouver une preuve du problème ou échouent dans le temps ou l'espace mémoire qui leur est imparti sinon.

Les outils à comparer sont en réalité un outil de déduction automatique et une extension de cet outil. Il s'agit de l'outil Zenon et de son extension Zenon Modulo. L'extension Zenon Modulo est censée améliorer l'outil Zenon, d'où l'intérêt d'une interface permettant d'exécuter ces deux outils et de les comparer en termes de nombre de problèmes automatiquement démontrés.

2 Cahier des charges

L'expression des besoins de ce projet est le suivant :

- L'utilisateur doit pouvoir d'identifier sur l'interface web avant de pouvoir l'utiliser. Un système de création de comptes doit donc être mis en place (en utilisant par exemple les adresses mails des utilisateurs). Il doit également y avoir un compte administrateur.
- L'utilisateur doit pouvoir sélectionner un ensemble de problèmes de la bibliothèque TPTP (voir la section 5 pour récupérer cette bibliothèque). Il faut uniquement considérer les problèmes du premier ordre (catégorie FOF), et pour les sélectionner, il faudra utiliser un outil spécifique qui vient avec la bibliothèque TPTP (outil tptp2T).
- L'utilisateur doit pouvoir appeler les outils Zenon et Zenon Modulo (ensemble ou pas) sur les problèmes sélectionnés (voir la section 5 pour récupérer ces outils). L'utilisateur doit pouvoir sélectionner

la limite en temps et en mémoire concernant l'exécution de ces outils (les deux outils prennent des options permettant de positionner ces limites).
- Pendant l'appel de ces outils, l'utilisateur doit pouvoir vérifier où l'exécution en est et faire afficher les problèmes déjà traités. L'affichage doit regrouper le numéro du problème, si les outils ont réussi à trouver une preuve et si oui en combien de temps.
- Après l'appel de ces outils sur un ensemble de problèmes donnés, l'interface doit afficher l'intégralité des résultats, ainsi que des statistiques sur cet appel. Parmi les statistiques, il doit y avoir le nombre total de problèmes considérés, le nombre de problèmes démontrés par un outil, le nombre de problèmes où ce même outil échoue, et dans le cas d'un appel simultané des deux outils, le nombre de problèmes démontrés par un outil et pas par l'autre et vice-versa.
- Après un appel de ces outils, l'utilisateur doit pouvoir enregistrer ces résultats dans une base de données. Les informations à enregistrer sont les suivantes : nom de l'utilisateur qui a fait le test, date, version de la bibliothèque TPTP utilisée, versions des outils utilisés, paramètres de l'appel (limites en temps et en mémoire), et problèmes testés avec les résultats obtenus par les outils (avec le temps mis pour démontrer un problème dans le cas d'un succès). Outre ces informations, l'utilisateur doit pouvoir également enregistrer dans cette base de données un texte libre (typiquement un commentaire) correspondant à cet appel.
- L'utilisateur doit pouvoir consulter des anciens résultats qui ont été stockés dans la base de données. Pour rechercher ces résultats, l'utilisateur doit pouvoir le faire par différents critères, comme le nom de l'utilisateur, la date du test, et tous les champs de la base de données pour lesquels cette recherche fait du sens. L'utilisateur doit pouvoir combiner la recherche sur ces différents critères, et pour plus de flexibilité, réaliser directement des requêtes SQL.
- L'utilisateur doit pouvoir consulter à tout moment un guide d'utilisation de l'interface, et des aides doivent être systématiquement présentes lorsque notamment des demandes de saisies sont réalisées par l'interface. Ces aides doivent être ergonomiques et contextuelles.
- L'interface devra être correctement présentée du point de vue du style. Pour ce faire, il n'y aucune limite dans l'utilisation d'outils appropriés (il est typiquement possible d'utiliser des outils qui génèrent automatiquement des feuilles de style CSS).
- Le code de l'interface devra être correctement commenté à tous les niveaux (HTML, PHP, ...). Des fichiers d'aide (fichiers README, INSTALL, ...) doivent être fournis pour faciliter la maintenance et l'installation de cette interface.
- Le code de l'interface doit pouvoir être « packagé » facilement et automatiquement (en une seule commande). Il doit pouvoir s'installer sur n'importe quelle architecture et système (même si les systèmes à la Unix seront privilégiés).

3 Extensions possibles

Dans le cas où tous les besoins du cahier des charges ont été satisfaits, il sera possible de considérer un certain nombre d'extensions (qui apporteront des points de bonus) :
- Une extension consisterait à pouvoir considérer plusieurs versions de Zenon et Zenon Modulo, et à pouvoir choisir les versions que l'on souhaite appeler. On pourrait également souhaiter pouvoir gérer plusieurs versions de la bibliothèque TPTP.
- Une autre extension consisterait à considérer d'autres outils de déduction automatique. Concernant le projet de recherche sur lequel ce projet s'appuie, il serait intéressant de considérer notamment les outils iProver et iProver Modulo.

4 Travail collaboratif

Ce projet devra être un travail de groupe. Un découpage fonctionnel du projet en tâches devra être au préalable réalisé (avant même de commencer quoique ce soit), et ce découpage fonctionnel devra être validé par l'ensemble du groupe. Une liste des tâches (précisément décrites) devra être réalisée et le lien entre ces différentes tâches devra également apparaître (pour ce faire, un schéma faisant apparaître ces dépendances entre tâches pourra avantageusement répondre à ce besoin). Un chef de projet devra être nommé et se chargera de « manager » le groupe (une tâche explicite de management devra également apparaître dans le découpage fonctionnel du projet). Le chef de projet sera également chargé de contrôler la présence de chacun aux différentes séances de travail (il pourra, par exemple, le faire au moyen d'une base de données qu'il créera au début du projet). Une bonne assiduité apportera un bonus à la note individuelle d'un auditeur, tandis que trop d'absences pénaliseront cette même note.

5 Adresses utiles

Compilateur OCaml : `http://caml.inria.fr/ocaml/`.
Outil Zenon : `http://focal.inria.fr/zenon/`.
Outil Zenon Modulo : `https://www.rocq.inria.fr/deducteam/ZenonModulo/`.
Bibliothèque TPTP : `http://www.tptp.org/`.

6 Dates et barème

Le projet dans son intégralité doit être committé sur l'archive Git de GitHub le 27 juin 2014 au plus tard. Les soutenances auront lieu les 27 juin 2014 et 3 juillet 2014. Elles seront constituées d'une soutenance globale et de plusieurs soutenances individuelles. Chaque auditeur obtiendra une note pour la soutenance globale (la même pour tout le monde), ainsi qu'une note pour sa soutenance individuelle. La note finale sera la moyenne de ces deux notes. En cas d'échec à la première session, la note globale sera conservée par l'auditeur, et seule la note individuelle pourra faire l'objet d'un rattrapage.