Résumé des discussions lors de la journée « bibliothèques »

Discussion sur les objectifs de la bibliothèque standard Coq :

- un aspect pédagogique en général : que la bibliothèque propose les définitions et propriétés nécessaires aux tutoriels sur Coq, qu'on puisse avoir accès à des structures simples et « standard » (t.q. entiers de Peano, Z, listes, propriétés algébriques de base vérifiées par ces structures, mais aussi réels ?)
- un aspect pédagogique pour l'informatique en particulier : que la bibliothèque propose les définitions de base pour travailler avec Coq comme un langage de programmation (Z et ses opérations, listes, strings, ...)
- support pour les tactiques du système
- une taille gérable par l'équipe de développement de Coq
- « standard »

Organisation

– une approche plus souple des contributions utilisateurs en tant que bibliothèques
– la mise en place d'une possibilité d'import dynamique de bibliothèques « vivantes » (e.g. CoRN, CompCert, CoLoR, ...)

Quels composants ?

– booléens
– listes
– relations, ordres bien fondés
– strings
– arithmétique (concret ; mais aussi modulaire ? jusqu'où ?)
– (rationnels ?)
– (réels ?)
– (fsets/fmaps ?)
– (tableaux ?)
– (complexes ?)

Pas de consensus sur les noms entre parenthèses.

Autres questions et remarques

– les bibliothèques doivent être « jolies »
– ne plus faire bouger les morceaux de la bibliothèque dont on est content ?

Quelles lignes directrices pour le style de preuve ? Plusieurs critères, pas vraiment de consensus ?

– souci du contenu calculatoire ?
– robustesse ?
– longueur du script ?
– place de l'automatisation ?
– raisonnement abstrait (p.ex. réécriture) ou calculatoire (p.ex. `simpl`) ?
– structuration, indentation

Éléments d'une ligne directrice pour la conception d'une bibliothèque standard

– contenu : mettre les lemmes qui font du sens même si leur preuve est esentiellement calculatoire (e.g. `0+n=n` sur `nat`)
– relations d'ordre : compte tenu des capacités actuelles de Coq, $>$ et $\geq$ en notation « only parsing » est *la* bonne solution

Autre décision : ne plus mettre les preuves dans la stdlib sur le web.

Quelques remarques relatives à l'exposé d'Assia

Assia souligne divers manques de la bibliothèque standard :

– manque de partage entre les différentes parties de la bibliothèque : e.g. pas de notion d'ensemble, pas de notion unique de polynôme,
– manque d'automatisation modulaire.

Assia montre des exemples de définitions calculatoires dans la bibliothèque ssreflect sur `nat` : une seule définition pour $\leq$, $<$, $\geq$, $>$. Montre des définitions de spécifications sur mesure à la `sumbool` mais dont les arguments sont booléens (donc « proof-irrelevant »).

Montre comment les structures canoniques et les coercions permettent de réutiliser du code, par exemple, les résultats sur les listes s'appliquent aux « listes de taille fixe » par projection et les résultats sur les listes de taille fixe aux listes usuelles par équipement canonique.

Quelques remarques relatives à l'exposé de Sylvie et Guillaume

Sylvie et Guillaume ont suggéré plusieurs améliorations de la bibliothèque `Reals` afin d'en améliorer la cohérence et l'utilisabilité.