Interactive Theorem Proving and Applications

Année 2022-23

Prof. Burkhart Wolff wolff@lri.fr

www.lri.fr/~wolff/teach-material/2022-2023/M2-CSMR/

TP 4 - Inductive Constructs in Isabelle

Semaine du 31 janvier 2023

Exercice 1 (Inductive sets - Inductive Proofs)

Define a (polymorphic) regular expression language $\alpha\ rexp$ with the alternatives :

- `Empty` (denoted <>)
- `Atom` (a singleton, denoted ⌊_⌋)
- `Alt` (for alternative, denoted _|_)
- `Conc` (for sequence, denoted _ : _)
- `Star` (for arbitrary repetition)

Tasks :

1. Why is $((A :: \alpha\ rexp)|B) = (B|A)$ not true in general ?
2. Define inductively : if A is a language, then $star\ A$ is the set of all repetitions over A.
3. Define recursively L, the language of a regular expression.
4. Prove $star\{\} = \{[]\}$ and therefore $star(star\{[]\}) = \{[]\}$.
5. Prove that L commutes over _|_.
6. Prove that under L, _ : _ distributes over _|_ (left and right).
7. Prove that the word `''acbc''` is in the language of $Star((\lfloor CHR''a''\rfloor|\lfloor CHR''b''\rfloor) : \lfloor CHR''c''\rfloor)$

Note : Main provides the notation `CHR ''a''` for "the character a". Strings are defined as lists of characters.

Exercice 2 (Parametric Inductive Sets)

1. Definir le prédicat `path` qui etabli s'il y a un chemin de deux points differents dans un graphe donné par une relation de type $(\alpha \times \alpha)set$.
   (Hint : Il s'agit d'une forme de cloture transitive non-reflexive).
2. Preuve : si la relation est just le successeur (+1), que qu'il y a un chemin de 11 a 13 dans cette relation.
3. Pour n'importe qu'elle relation A, s'il y a un chemin de `a` a `b`, et s'il y a un chemin de `b` a `c`, il y a un chemin de `a` a `c` (transitivité).

Hints :

1. apply the variant proof method : `induct` *if possible* !

Exercice 3 (Modelling and Proof : The typed λ-calculus)

Define the λ-calculus as a data-type inside HOL. (This is also called a "deep embeding" into HOL). The first 3 parts are identical to TP 3.2.

1. Define the “terms” (abstract syntax tree) of the untyped λ-calcul as “data type”
2. Define the “types” (abstract syntax tree) du λ-calcul as “data type”
3. Define a function `instantiate` for that substitutes type-variables against types.
4. The environments Σ et Γ by using association lists.
5. Define inductively the well-typedness quartuple : a term t is well-typed with type τ in the environnements Σ et Γ.
6. Define a Σ_0 with the constants True, False, and equality inside our λ-calculus model.
7. Prove that in Σ_0 the encoding of the term $(\_ = \_)(True)$ has the (encoding of) the type *bool* $\rightarrow$ *bool*.
8. Define Σ according to slide 30 in the module "U1 - λ-calculus" and prove that $(\_ = \_)(\_ = \_)$ is typeable in Σ.

Exercice 4 (OPTIONAL : Report)
(IN CASE THAT YOU WANT TO HAVE IT GRADED. RECALL THAT 2 OUT OF 6 TP’s SHOULD BE SUBMITTED.)

1. Write a little report answering all questions above, note the difficulties you met, add some screenshots if appropriate. 3 pages max (except screenshots and other figures).