Prof. Burkhart Wolff
wolff@lri.fr

TP 6 - Modeling Operational Semantics

Semaine du 8 fevrier 2021

Objective : Defining an operational semantics "Plotking-style" big-step semantic for as more-or-less standard regular expression language. This type of semantics represents "states" $'s$ in the original language and inductively models the transition relation between states via a transition predicate $\langle \_, \_ \rangle \longrightarrow_c \_$ of type $'s \times' output\ option \times' s \Rightarrow bool$ where *option* is the usual option type constructor from `Main`. Following common terminology in automata theory, we will call a list of atoms a *word*, and a set of words a *language*. The *denotational semantics* we are referring here is a function that maps an (abstract) *syntax* to a set of *deonotations*, i.e. regular expressions to the language they denote. We will introduce the term *epsilon* as abbreviation for *Star Empty*, an $\alpha\ rexp$ term for the empty language.

We reuse the abstract syntax $\alpha\ rexp$ of the regular expression language from TP5.

Exercice 1 (Inductive Sets - Inductive Proofs)
Define a Plotkin-style semantics for regular expressions, where $'output$ is set to $'\alpha$. Complete the list of inductive rules starting with :

— $\langle \lfloor a \rfloor, Some\ a \rangle \longrightarrow_c\ epsilon$
— $\langle Empty : R, None \rangle \longrightarrow_c\ Empty$
— $\langle \lfloor a \rfloor : R, Some\ a \rangle \longrightarrow_c\ R$
— $\langle Star\ r, None \rangle \longrightarrow_c (r : (Star\ r))$
— $\langle Star\ r, None \rangle \longrightarrow_c epsilon$
— ...

Tasks :

1. Prove $\langle \lfloor a \rfloor, Some\ a' \rangle \longrightarrow_c epsilon = (a' = a)$ and $\langle \lfloor a \rfloor, a' \rangle \longrightarrow_c R' = (a' = Some\ a \wedge R' = epsilon)$ (This should hold for your completion of the above inductive rule-set).
2. Derive all similar lemmas resulting from your definitions (should be approx 8). Hint : for the latter rule, there is actually a specific command that derives this type of simplification lemmas automatically. For example, the last mentioned lemma could be derived automatically via :
$$inductive\_simps\ atom1S\ :\ "\langle \lfloor a \rfloor, a' \rangle \longrightarrow_c R'"$$
3. Prove the eliminiation rule :
$$\langle \lfloor a \rfloor, Some\ a' \rangle \longrightarrow_c epsilon \Longrightarrow (a' = a \Longrightarrow P) \Longrightarrow P$$
4. Prove all other elimination rules and configure them into the global context as such. Hint : for the latter rule, there is actually a specific command that derives this type of simplification lemmas automatically. For example, the last mentioned lemma could be derived automatically via :
$$inductive\_simps\ atom1S\ :\ "\langle \lfloor a \rfloor, a' \rangle \longrightarrow_c R'"$$

5. Now define the mu ;tiple step semantics Plotkin style. This hsould lopok like this :

```
inductive
  evalstar ::        "['a rexp,'a list,'a rexp] ⇒ bool" ("⟨_,_⟩/ ⟶c* _" [0,0,60] 60)
where
  idle:              "⟨epsilon,[]⟩ ⟶c* epsilon"
| step1:             "⟨r,Some a⟩ ⟶c r' ⟹ ⟨r,[a]⟩ ⟶c* r'"
| continuation1:     "⟨r,None⟩ ⟶c r'   ⟹  ⟨r',S⟩ ⟶c* r'' ⟹ ⟨r,S⟩ ⟶c* r''"
| continuation2:     "⟨r,Some a⟩ ⟶c r' ⟹  ⟨r',S⟩ ⟶c* r'' ⟹ ⟨r,a#S⟩ ⟶c* r''"
```

6. Prove :

$$\langle Star((\lfloor CHR''a''\rfloor | \lfloor CHR''b''\rfloor) : \lfloor CHR''c''\rfloor), ''bc''\rangle \longrightarrow_c^* epsilon$$

7. Prove :

```
theorem operational_implies_denotational_generalized':
assumes nat_steps: "(⟨r,s⟩ ⟶c* r')"
    and den_cont: "∃as. as ∈ L r''"
shows   "∃ as . s@as ∈ L r ∧ as ∈ L r''"
```

8. Prove the main theorem "operational semantics implies denotational semantics" :

$$(\langle r, s\rangle \longrightarrow_c^* epsilon) \longrightarrow s \in L(r)$$

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

Exercice 2 (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).