Gramáticas

1 Gramáticas

```
v, name \in Var = String
                                            d \in RawValue ::= nil \mid d \cdot d'
                                                                                                 Value := d \mid p
p \in Prog ::= \mathbf{proc} \text{ name : bp}
                 | proc name : xp
e \in Expr ::= v \mid \mathbf{nil} \mid \mathbf{cons} \; e \; e'
                  | hd e | tl e | =? e e'
                  | * | univ e e'
c\in Comm::=v:=e
                   | while e do c od
                   | c; c'
bp \in BaseProg ::= read \ v \ ; c \ ; write \ w \ ;
pat \in Pattern ::= nil \mid pat \cdot pat' \mid v
xc \in ExtComm ::= v := e
                           | v := name v'
                             \mathbf{while} \, \in \mathbf{do} \, \, \mathbf{xc} \, \, \mathbf{od} \,
                             xc; xc'
                             if e then xc fi
                             if e then xc else xc' fi
                             rewrite [v_1, \ldots, v_n] by [\operatorname{pat}_1^1, \ldots, \operatorname{pat}_n^1] ==> (\operatorname{xc} \mid [e_1^1, \ldots, e_n^1])
                                  [\operatorname{pat}_1^m, \ldots, \operatorname{pat}_n^m] \stackrel{\cdot}{=} > (\operatorname{xc} \mid [\operatorname{e}_1^m, \ldots, \operatorname{e}_n^m])
                              end
                            pat^{1} ==> xc^{1}
\vdots
pat^{m} ==> xc^{m}
end
                            case v of
xp \in ExtProg ::= \mathbf{read} \ v \ ; xc \ ; \mathbf{write} \ w \ ;
```