SDE and automaton

Marco Beccuti

Università degli Studi di Torino

Dipartimento di Informatica

November 2014

Automaton grammar

⟨*Automaton*⟩ ::= { ⟨*States*⟩ } ; **ENDL** ⟨*Graph*⟩ |
% ⟨*Text*⟩ **ENDL** ⟨*Automaton*⟩ | **%** ⟨*Text*⟩ **ENDL**

⟨*States*⟩ ::= **STATE** | **STATE** ⟨*Spec*⟩ |
STATE ⟨*Spec*⟩ , ⟨*States*⟩ | **STATE** , ⟨*States*⟩

⟨*Graph*⟩ ::= ⟨*Node*⟩ ; **ENDL** |
⟨*Node*⟩ ; **ENDL** ⟨*Graph*⟩| **ENDL**⟨*Graph*⟩ | **ENDL**

⟨*Node*⟩ ::= **STATE (** ⟨*Condition*⟩ **)** | **STATE (** ⟨*Condition*⟩ **)**{ Transitions }
STATE { Transitions }

⟨*Transition*⟩ ::= ⟨*Condition*⟩ -> **STATE** |
⟨*Condition*⟩ -> **STATE ;** ⟨*Transition*⟩

⟨*Condition*⟩ ::= ⟨*Exp*⟩ < ⟨*Exp*⟩ | ⟨*Exp*⟩ > ⟨*Exp*⟩ | ⟨*Exp*⟩ <> ⟨*Exp*⟩ |
⟨*Condition*⟩ || ⟨*Condition*⟩ | ⟨*Condition*⟩ **&&** ⟨*Condition*⟩ |
! ⟨*Condition*⟩ | **(** ⟨*Condition*⟩ **)**

⟨*Exp*⟩ ::= ⟨*Exp*⟩ ∗ ⟨*Exp*⟩ | ⟨*Exp*⟩ \ ⟨*Exp*⟩ |
⟨*Exp*⟩ + ⟨*Exp*⟩ | ⟨*Exp*⟩ − ⟨*Exp*⟩ |
STRING | **$T** | **NUMBER** |

⟨*Text*⟩ ::= ⟨*Text*⟩ **STRING** | **STRING**

⟨*Spec*⟩ ::= **@** | **#**