Appendix A – Syntactic symbols used in Jai

## " " delimiting a string literal

## \ special characters

## % substitution of values (like in print) (the insert symbol):

. operator for both struct membership and pointer dereference access

## .. range notation 0..n-1 / definition of dynamic arrays [..] /

## .. variable number of arguments in a function

## ; end of code line / separating members of an enum

## : type declaration / separate counter from range in for-loop

## = initialization, assignment

## := declaration with type inference

## == test on value equality

## != test on value inequality

## ! not

## < <= > >=

## --- uninitialized variable

## , separating parameters in parameter list of a function

## :: definition of constants, types, structs, functions,

## ( ) to indicate operator priority / function syntax: parameter list of a function , calling a function with arguments

## { } encloses any code block / definition of structs

## [ ] definition of arrays

## -> return type of a function

## \* pointer, placed to the left of variable name / address of operator

## $ parameter in polymorphic procedure

## << dereference a pointer

## + sum operations n += 1

## \* /

## += -= \*= /\*

\*~ relative pointer

/ polymorph interface (see § 23.4)