Specification of Red JonPRL Signatures

Jon Sterling

January 10, 2016

1 Grammar

The grammar of Red JonPRL signatures is presented in Figure 1. Note that an optional production of sort s is formatted $\langle s \rangle$ in the rules.

sigexp	::=	$\langle \cdot \rangle$	empty signature
		sigexp. *sigdec*	signature extension
sigdec	::=	`Def` *opid* $\langle [sctx] \rangle$ $\langle (mctx) \rangle$: *sortid* = [*term*]	operator definition
		`Tac` *opid* $\langle [sctx] \rangle$ $\langle (mctx) \rangle$ = [*tac*]	tactic definition
		`Thm` *opid* $\langle [sctx] \rangle$ $\langle (mctx) \rangle$: [*term*] `by` [*tac*]	theorem declaration
sctx	::=	$\langle \cdot \rangle$	empty symbol context
		sctx, *symbind*	symbol context extension
mctx	::=	$\langle \cdot \rangle$	empty metavariable context
		mctx, *metabind*	metavariable context extension
symbind	::=	*symid* : *sortid*	symbol binding
metabind	::=	*metaid* : *valence*	metavariable binding
valence	::=	$\langle \langle \{ sortlist \} \rangle \langle [sortlist] \rangle . \rangle$ *sortid*	valence
sortlist	::=	$\langle \cdot \rangle$	empty sort list
		sortlist, *sortid*	sort list extension

Figure 1: Grammar of signature expressions. The identifier sorts *opid*, *sortid*, *symid* and *metaid* can be assumed to be arbitrary strings; the sorts *term* and *tac* are left uninterpreted.