This is a specification of the subset of Standard ML supported by this compiler. Note that every program valid in this subset will be valid in SML. Semantics are preserved.

atexp	:=	*con*	constant(int, bool), like **1**, **true**
		vid	variable identifier
		(*exp*)	parenthesized
exp	:=	*atexp*	atomic
		exp atexp	application
		exp_1 *binop* exp_2	infix application
		fn (<*vid* **:** *typ*>$^{(,)}$) => *exp*	anonymous function
		if exp_b **then** exp_t **else** exp_f	if expression
dec	:=	**val** *vid* **:** *typ* = *exp*	value bind
attyp	:=	*tid*	type variable
		(*typ*)	parenthesized
typ	:=	*attyp*	atomic
		typ <∗ *typ*>$^{+}$	tuple type
		typ_1 -> typ_2	arrow type

Supported built-in types: **int**, **bool**, **unit**.

Not supporting declaration of operators as infix.

No identifiers from the bare language can be rebound.

fn arguments must be surrounded by parentheses, and must be a (potentially empty) comma-separated list of value identifiers.

fn arguments need to be type annotated.

Type identifiers that start with **'**, such as **'a**, are polymorphic.