### Scheme Quick Reference

**Constructing & Manipulating Lists:**
- (cons arg1 arg2) E.g. (cons '(a) '(b c d)) result: ((a) b c d)
- (append arg1 arg2) E.g. (append '(a) '(b c d)) result: (a b c d)
- (list arg1 arg2...) E.g. (list 'a 'b '(c d)) result: (a b (c d))

**Conditional & Selection Statements:**
- (let ( (<var1> <exp1>) ...
  ... ) )

**Functional Abstraction**
- (lambda <expr-list> <body>)

---

### ML Quick Reference

**Lists:**
- [obj1,obj2,...] @ operator E.g. [1,2]@[3] result: [1,2,3]
- :: operator E.g. 1::[2] result: [1,2]

**Tuples:**
- (obj1, obj2,...)

**Functions:**
- fun <func-name> <input-param> = <expression>;
- fn <func-param> => <func-body>;
- fun <func-name> <pattern1> = <expression1>
  |  <func-name> <pattern2> = <expression2>
  ...
  |  <func-name> <patternn> = <expressionn>;

**Conditional & Selection Statements:**
- (let ( (<var1> <exp1>) ...
  ... ) )

**Functional Abstraction**
- (lambda <expr-list> <body>)

---

### Prolog Quick Reference

**Horn clause:**
- <head> :- <body>

**Cut Operator:**
- ! operator

**Lists:**
- [] Empty list
- | operator E.g. [H|T]=[a,b,c] means H=a, T=[b,c]

**Function terms:**
- functor(parameter1,parameter2,...)

**Boolean Predicates:**
- X = Y Succeeds if X and Y can be unified.
- X \= Y Succeeds if X and Y cannot be unified.
- X == Y Succeeds if X and Y are already instantiated to the same object.
- X <= Y Succeeds if X and Y are not already instantiated to the same object.
- X =:= Y Succeeds if X and Y are identical after evaluating both terms.
- X is Expr Evaluate Expr and unify with X.

**Logical Operators:**
- "+" not provable
- , logical conjunction (AND)
- ; logical disjunction (OR)

---

### Prolog Quick Reference

**Horn clause:**
- <head> :- <body>

**Cut Operator:**
- ! operator

**Lists:**
- [] Empty list
- | operator E.g. [H|T]=[a,b,c] means H=a, T=[b,c]

**Function terms:**
- functor(parameter1,parameter2,...)

**Boolean Predicates:**
- X = Y Succeeds if X and Y can be unified.
- X \= Y Succeeds if X and Y cannot be unified.
- X == Y Succeeds if X and Y are already instantiated to the same object.
- X <= Y Succeeds if X and Y are not already instantiated to the same object.
- X =:= Y Succeeds if X and Y are identical after evaluating both terms.
- X is Expr Evaluate Expr and unify with X.

**Logical Operators:**
- "+" not provable
- , logical conjunction (AND)
- ; logical disjunction (OR)