1. **UNIT COLOUR SETS**

**Declaration Syntax**

```
colset name = unit [with new_unit];
```

**Declaration Examples**

```colset U = unit;
colset E = unit with e;```

2. **BOOLEAN COLOUR SETS**

**Declaration Syntax**

```
colset name = bool [with (new_false, new_true)];
```

**Declaration Examples**

```colset B = bool;
colset Answer = bool with (no, yes);```

**Boolean Operations**

- **Not** `b`  
  negation of the boolean value `b`
- `b1 andalso b2`  
  boolean conjunction, and
- `b1 orelse b2`  
  boolean disjunction, inclusive or

3. **INTEGER COLOUR SETS**

Integers are numerals without a decimal point. The integer colour set is large unless restricted by the `with` clause, in which case it is small.
### Declaration Syntax

```
colset name = int [with int-exp1...int-exp2];
```

### Declaration Examples

```
colset INT = int;

colset SmallInt = int with 1..10;
```

### Operations

- `~i`: negation of the integer value `i`
- `i1 + i2`: addition
- `i1 - i2`: subtraction
- `i1 * i2`: multiplication
- `i1 div i2`: division, quotient
- `i1 mod i2`: modulus, remainder
- `abs i`: absolute value of `i`
- `Int.min(i1,i2)`: minimum of `i1` and `i2`
- `Int.max(i1,i2)`: maximum of `i1` and `i2`

### 4. STRING COLOUR SETS

Strings are specified by sequences of printable ASCII characters surrounded with double quotes. The string colour set is large unless restricted by the `with ... and` clause, in which case it is small.

### Declaration Syntax

```
colset name = string [with string-exp1..string-exp2 [and int-exp1..int-exp2]];
```

### Declaration Examples

```
colset S = string;

colset LowerString = with "a".."z";

var lowerString : LowerString;
```

A string colour set is used in the example CP-net Simple Protocol.
Operations

\[ s_1^s_2 \]  concatenate the strings \( s_1 \) and \( s_2 \)

**String.size**  \( s \)  number of characters in \( s \)

**substring**  \((s, i, \text{len})\)  extract a substring of length \( \text{len} \) starting at position \( i \) in \( s \), first position is 0

**explode**  \( s \)  convert string \( s \) to list of chars

**implode**  \( l \)  convert list \( l \) of chars to a string

5. **ENUMERATED COLOUR SETS**

Declaration Syntax

```plaintext
colset name = with id0 | id1 | ... | idn;
```

Declaration Examples

```plaintext
colset Day = with Mon | Tues | Wed | Thurs | Fri | Sat | Sun;
```

6. **INDEX COLOUR SETS**

Declaration Syntax

```plaintext
colset name = index id with int-exp1..int-exp2;
```

Declaration Examples

```plaintext
colset PH = index ph with 1..5;
```

**PRODUCT COLOR SETS**

Declaration Syntax

```plaintext
colset name = product name1 * name2 * ... * namen;
```

where \( n \geq 2 \)

Declaration Examples

```plaintext
colset P = product U * I;
```