

Sample programs for JSetL

A. General (simple) programs (package `samplePrograms.gen`)

1. `AllCombinations_n_k` - Compute all combinations of k distinct elements out of n elements
2. `AllPermutations_n_using_in` - Compute all permutations of $1..n$
3. `Permutations` - Compute all permutations of $1..n$, one by one
4. `SortList_using_FD` - Sort a set of n integers
5. `AllPairs` - Compute the set of all ordered pairs $[x,y]$ s.t., $x \neq y$
6. `AllUnorderedPairs` - Compute the set of all unordered pairs $[x,y]$ s.t., $x \neq y$
7. `Concat` - Concatenate two lists (at least one known)
8. `Prefix` - Check whether a list is a prefix of another (both known, using `concat`)
9. `Max_Min` - Compute the maximum/minimum of a set of integers (declarative solution)
10. `Squares_1_n` - Compute squares
11. `Divisor` - Operations on numbers that involve the divisors, like printing all divisors of a number or checking whether the number is prime or not
12. `CartesianProduct` - Computes the cartesian product of two closed sets
13. `FilteredCartesianProduct` - Computes the cartesian product of two closed sets and filters it with constraints on the first and second component of the resulting pairs
14. `PowerSet` - Computes the powerset of a closed set
15. `AllTriplesUsing_setof` - Collecting all solutions using `setof`: all triples
16. `AllSetUnificationUsing_setof` - Collecting all solutions using `setof`
17. `SeasonsTranslate` - Translate season names

B. Programs dealing with graphs (package `samplePrograms.graphs`)

1. `Tsp` - Travelling Salesman Problem
2. `TspWeighted` - Travelling Salesman Problem: weighted
3. `ReachableNodes` - Computes the set of reachable nodes in a graph (from a starting node)
4. `GraphTransitiveClosure` - Computes the reflexive transitive closure of a graph
5. `MapColoring` - Implements a solution of the map coloring problem
6. `MapColoringUSA` - Uses the solution of the map coloring problem to color a portion of the USA map
7. `Bipartite` - Bipartite graph (using map coloring FD)

C. Nondeterministic programs using user-defined constraints (package `samplePrograms.nd`)

1. `AllSublists` - Compute all sublists of a given list (using `jsetl.lib.LListOps`)
2. `SplitLList_using_concat` - Split a list into two sublists: all solutions (using `jsetl.lib.LListOps`)
3. `SplitListOfStrings_with_equal_length` - Split a list of strings into two sublists with the same total string length (using `jsetl.lib.LListOps`)
4. `FindPath` - Find a path in a given graph (using `jsetl.lib.LGraphOps`)
5. `FindPathMultipleUses` - Find all destinations/sources/graphs (using `jsetl.lib.LGraphOps`)
6. `ArithmExprParser` - A parser for simple arithmetic expressions (using DCG)

D. Programs using Restricted Intensional Sets (RIS) (package `samplePrograms.ris`)

1. `MinValue` - computes the minimum of a set of integers

2. Divisors - computes the set of divisors of a number
3. PrimalityTest - A (brute force) primality test
4. ListSum - computes the sum of the elements of a list
5. MapListSquares - updates an array of integers using function Square (higher-order function)
6. Median - computes the median of a set
7. CartesianProduct - computes the cartesian product of two sets
8. PowerSet - Computes the powerset of a set
9. PrimesSet - Computes the set of prime numbers up to a given number
10. SetOfSquares - Computes the set of squares of all even numbers in a given interval
11. MapColoringRIS - Implements a solution of the map coloring problem using RIS
12. Factorial - Computes the factorial of a number using RIS
13. ReachableNodes - Computes the set of reachable nodes in a graph using RIS
14. AckermannFunction - Ackermann function
15. MutuallyRecursiveFunctions - An example of mutually recursive functions using RIS
16. TheoremProving - checks various properties of sets and integer numbers using RIS
17. Grep -

E. Constraint Satisfaction Problems using FD or FS constraints (package samplePrograms.csp)

1. SendMoreMoney - SEND + MORE = MONEY puzzle
2. Coloring_using_FD - Implements a solution of the map coloring problem using FD constraints
3. Queens - n-queens
4. Sudoku - Sudoku
5. TernarySteiner - Steiner triple Systems
6. SocialGolfers - Social golfer problem
7. AcademicCurriculum - Academic curriculum
8. RadioLinkFreq - Radio Link Frequency