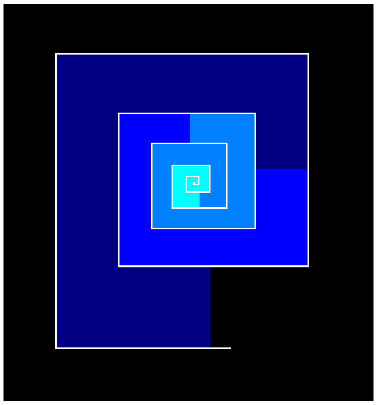
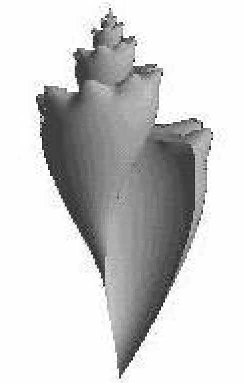
An Algorithms Guide

A brief guide to Algorithms Edited by Juan Chamero, jach_spain@yahoo.es, as of April 2006



A Darwin Search Agent progression strategy

Algorithms understanding and design is more than a game, well sorry, it could be in some extent considered the game of life, the intrinsic intelligence of nature to create the mystery of the old Taoist “Ten Thousand Things”. Given an algorithm running it forward is creation of beings that evolve sometimes from “seeds” sometimes from other beings, belonging to the same “family”. As time folds onward it may be most times unfolded backwards either pseudo random or deterministic. My passion by algorithms began reading The Growth and Form, the milestone work of D’Arcy Thompson, for many the first bio mathematician, when I was a teenager. His ideas about gnomes and gnomic transformations were in my humble opinion precursor of fractals.



Lyria

Extracted from “Seashell Surfaces”, G. Lucca, Itlay

Note: D’Arcy W. Thompson depicted the gnomic properties of equiangular spiral in seashells along the growth process. Seashells may be considered a connection point among geometry, mathematics and art. Since ancient times, seashells have attracted our attention because their beauty, elegance and symmetry.

Selected Authorities

We show here our suggested Basic Websites List to initiate your algorithmic navigation either for pleasure or deepening in the art of their understanding and design.

1. Math Forum, a well known math forum where you may find basic answers and guides to your questions: http://mathforum.org/library/drmath/sets/college_algorithms.html. You may find here answers to problems at elementary School, Middle School, High School, and College & Beyond levels. See as an example the meaningful answer about The Euclidean Algorithm: http://mathforum.org/library/drmath/view/51447.html.

2. Dictionary of Algorithms and Data Structures
National Institute of Standards and Technology, US
http://www.nist.gov/dads/
The Dictionary of Algorithms and Data Structures is an obliged reference for many of algorithms, "algorithmic techniques", "archetypal problems" and data structures. It is maintained by Paul E. Black, and is hosted by the Software Quality Group of the Software Diagnostics and Conformance Testing Division, Information Technology Laboratory, a part of the National Institute of Standards and Technology.

3. Example of use: Dijkstra Algorithm => (http://www.nist.gov/dads/HTML/dijkstraalgo.html) => an authority (http://www.cs.auckland.ac.nz/software/AlgAnim/dijkstra.html)

4. Data Structures and Algorithms
Computational Geometry Lab, from McGill University, http://www.mcgill.ca/
http://cgm.cs.mcgill.ca/~godfried/teaching/algorithms-web.html
See as an example Introduction to Trees, Class Notes, Topic#8:
http://www.cs.mcgill.ca/~cs251/OldCourses/1997/topic8/

5. Educational Individual Authority, Mark Allen Weiss, from FIU, Florida International University, US: http://www.cs.fiu.edu/~weiss/

6. The Algorithm Design Manual, http://www2.toki.or.id/book/AlgDesignManual/, a generous open Web contribution, better than a book!, by Steven S. Skiena, Department of Computer Science State University of New York, Stony Brook, NY 11794-4400. As an example take a look at http://www2.toki.or.id/book/AlgDesignManual/LEC/LECTURES/ALL.HTM, CSE 373/548 – “Analysis of Algorithms”.

7. Example of a well documented algorithm:
https://www.cs.auckland.ac.nz/software/AlgAnim/ppt/dijkstraC.ppt.gz, depict in a Power Point slides series a program stepwise of the Dijkstra Algorithm.

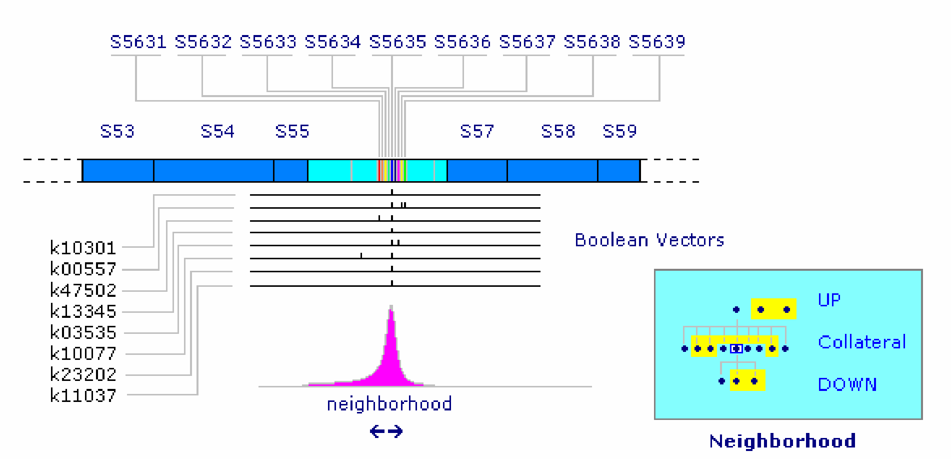
Algorithmic Images Gallery

Juan Chamero, jach_spain@yahoo.es, as of April 2006,
Extracted from works performed under his Direction

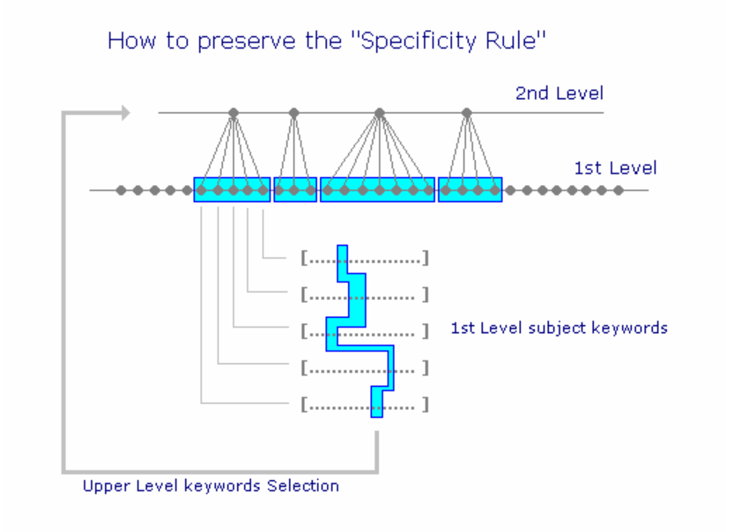
I) About Knowledge Management Applications

Here we show a series of images pertaining to Darwin, an AI multi-agent methodology oriented to unveil the "hidden" order of huge Data Reservoirs as the Web. Most algorithms work on High Order Sparse Binary Matrices defined over a GF2 three-dimensional space: 10 billionx10 millionx500,000.

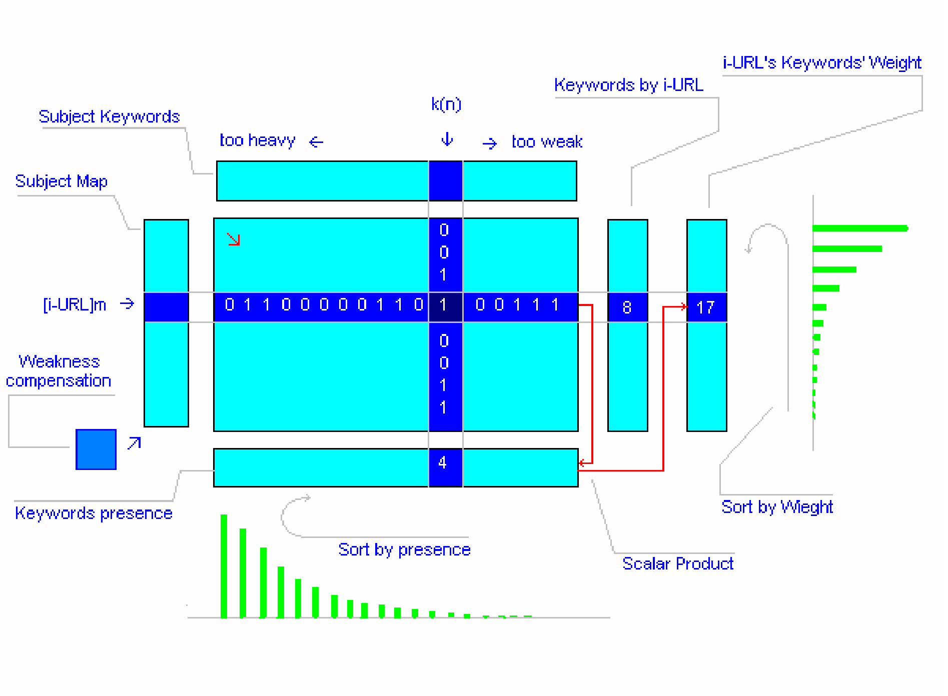
Mapping the Human Knowledge



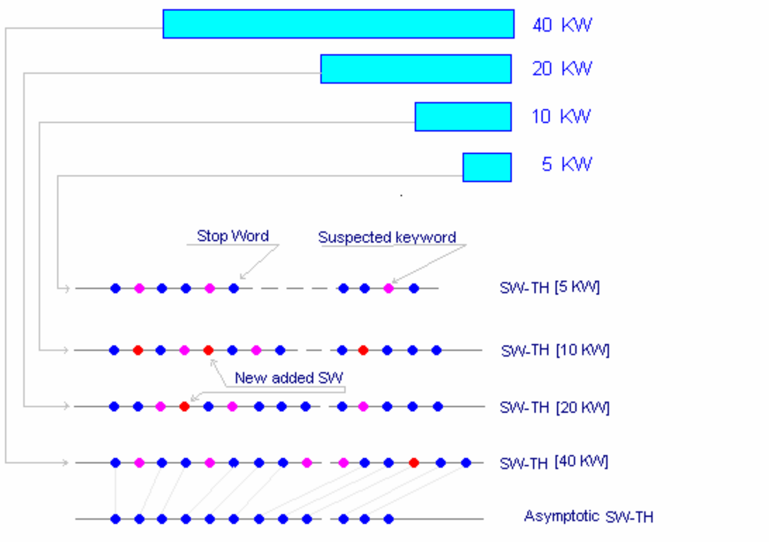
This figure depicts a complex multi agent task: mapping the knowledge of a huge Data Reservoir (for instance the Web). Resembling a DNA algorithm agents retrieve and discriminate all meaningful "concepts" out from million of documents of a given discipline by detecting clusters of them. Along its semantic skeleton they inspect the semantic neighborhood of any subject. See it at www.intag.org.


How to preserve the "Specificity Rule"
2nd Level
1st Level
1st Level subject keywords
Upper Level keywords Selection

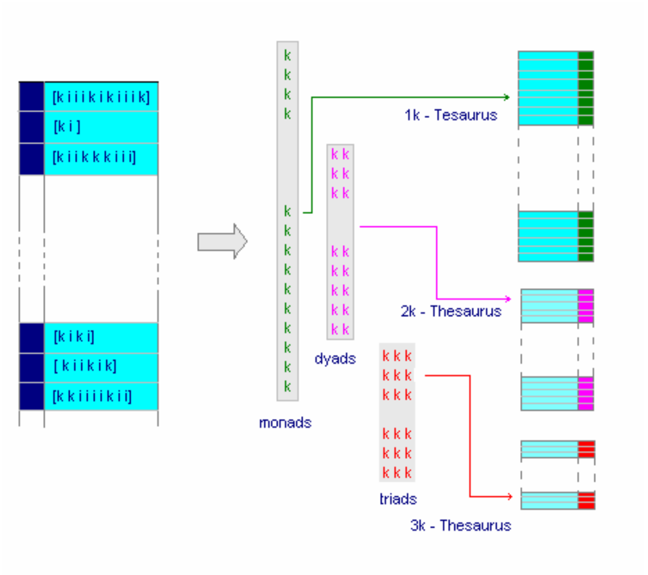
Detail of a sub algorithm performed by agents in the figure above. Given a "flat" unstructured Thesaurus (or Catalog) agents are settled and tuned up to build autonomously and automatically
the "hidden" hierarchy, level by level going upwards. Agents match make keywords versus subjects preserving the "Specificity Rule" a sort of entropy preservation.



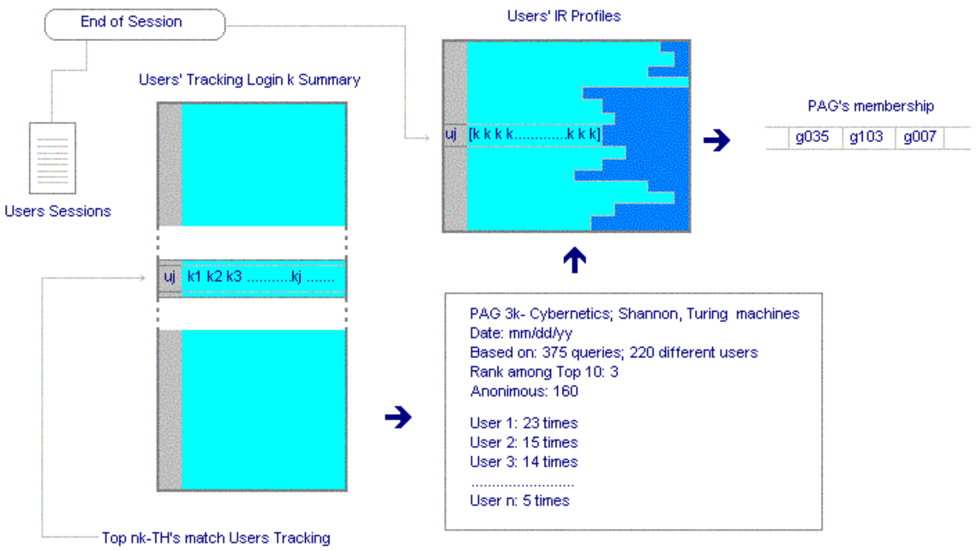
This algorithm work on one of these GF2 matrices building weighed spectrum of rather “exceptional” sequences of words/symbols, candidates to become concepts belonging to a given Major Subject mapped. .



This is another discrimination algorithm working on pure txt data samples, supposedly “written” with a common Jargon, growing exponentially until all that remains should be the Jargon itself.



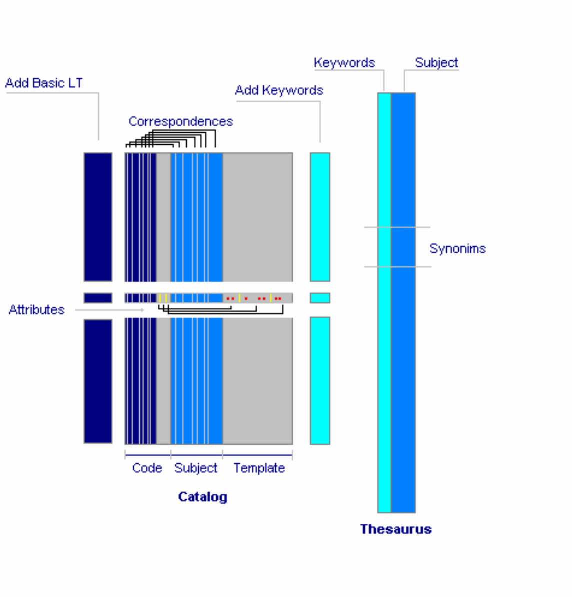
This algorithm works on “users’ side” trying to detect most frequent users’ behaviors/traits. Supposedly users intent to reduce their “uncertainty level” about something querying a given Cognitive Offer (for example via a Search Engine). Their queries are instrumented as “session strings” of guessed “keywords” k and navigation instances, all symbols detected through a “Darwin e-membrane” ®. Agents working on the “ectoderm” side of it discriminate potential “users’ keywords” (not the “established” keywords of the queried Knowledge Database) in monads, dyads, triads and n-ads, trying to build versions of a Users’ Thesaurus.


End of Session
Users' IR Profiles
Users' Tracking Login k Summary
Users Sessions
uj [k k k k.............k k k]
PAG's membership
g035
g103
g007
uj k1 k2 k3 ....kj ....
Top nk-TH's match Users Tracking

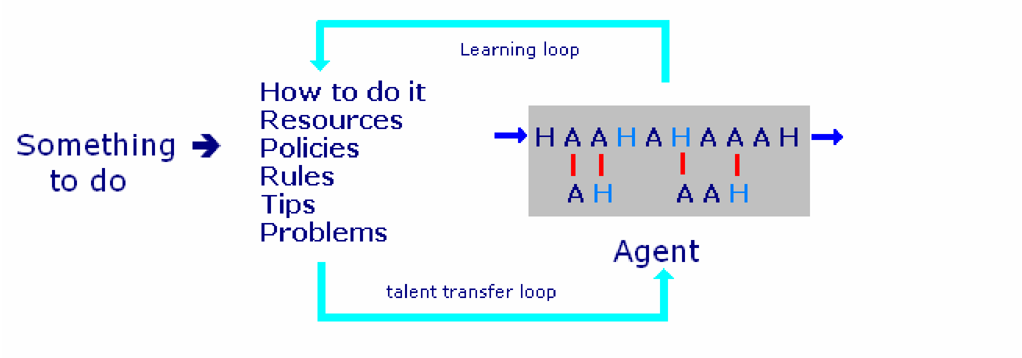
PAG 3k- Cybernetics; Shannon, Turing machines
Date: mm/dd/yy
Based on: 375 queries; 220 different users
Rank among Top 10: 3
Anonimous: 160

User 1: 23 times
User 2: 15 times
User 3: 14 times
....
User n: 5 times

Within a multi-agent platform, a bp-bot, behavior pattern agent, classify users’ sessions within pre-established PAG’s “groups”, characterized by very specific subjects semantic “Darwin fingerprints” ®, maintaining existing and suggesting new ones.



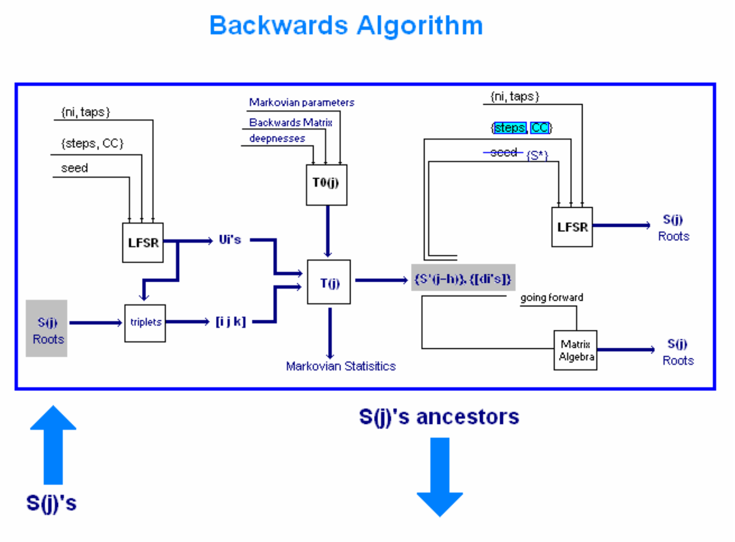
This figure shows the work of a Catalog Items procurement agent, p-bot, within a multi agent platform, performing some basic Catalog Harmonization tasks. Semantically it builds a Thesaurus with a "flat", unstructured only one level Catalog in as many levels as intrinsically exist. See it working at www.procurebot.com. Agents may work on both sides: classically, on "proprietary" side, and on "users' side", trying to understand and aid users daily operations.



This is the scheme of an "anthropic" multi-agent platform managed by KM Darwin Methodology ®. A given task "something to do" is solved via an anthropic Cosmo vision, "half humans", "half agents", doing each one what perform best. The purpose of the learning cycle is to transfer to agents as much as possible and as soon as possible, transferring "mechanical" talent to agents. We have experimented with Darwin and our conclusion is that what's computable, no matter its level of complexity could be successfully transferred to agents. Of course the ultimate control remains always in Human hands!.

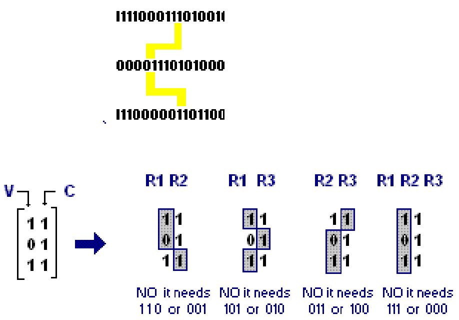
II) About a Telecom Application

Here we show a series of algorithms working on “flow streams” of digitalized signals, that have to be coded-decoded and errors detected-corrected along their way from origins to destinations. Some of these streams experiment hundreds of transformations via “states’ machines” like the one depicted below with five algorithms working synchronically under the control of a “main” program.



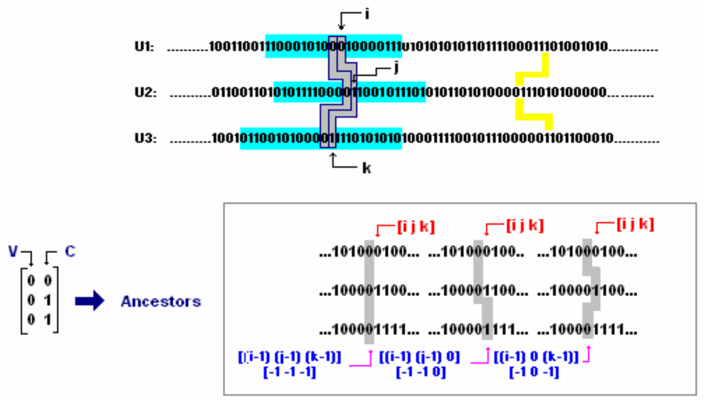
This algorithm deals with a rather complex backwards navigation throughout a “states machine”, specifically one based on a set of Non linear feedback registers. The core of the algorithm is a Markovian “critical” process.

Backwards null



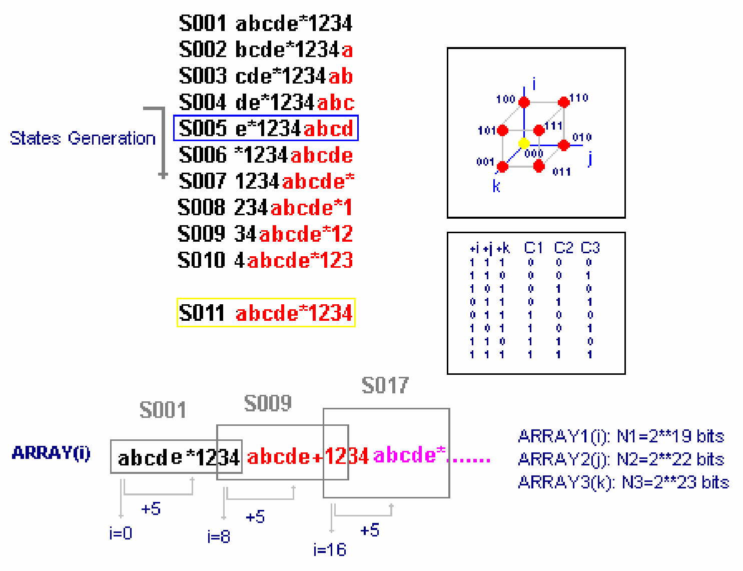
This algorithm shows a detail of the algorithm depicted above: navigating “upwards” a tree, the process when a node has no “ancestors”. It is also shown how the tree is browsed via a special “triplet” sub algorithm.

Navigation by triplets



This algorithm belongs to a higher level of the one shown above. Instead of browsing a 3D “states machine” binary space via “words arrays” performing it via “bits arrays”. Each “triplet” identifies state sequences either going forwards or backwards in the most efficient way.

Triplets Generation & Navigation



S001 abcde*1234
S002 bcde*1234a
S003 cde*1234ab
S004 de*1234abc
S005 e*1234abcd
S006 *1234abcde
S007 1234abcde*
S008 234abcde*1
S009 34abcde*12
S010 4abcde*123

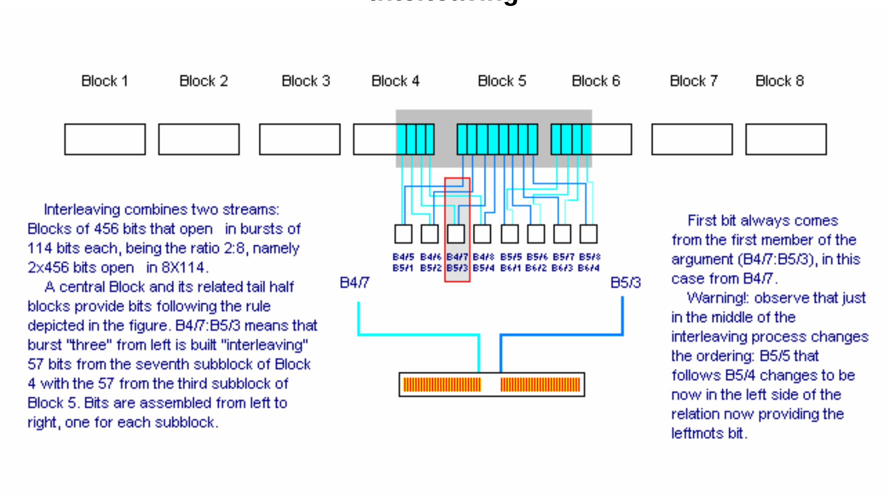
S011 abcde*1234

+i	+j	+k	C1	C2	C3
1	1	1	0	0	0
1	1	0	0	0	1
1	0	1	0	1	0
0	1	1	0	1	1
0	1	1	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	1	1	1

ARRAY1(i): N1=2**19 bits
ARRAY2(j): N2=2**22 bits
ARRAY3(k): N3=2**23 bits

We show here a zoom of above mentioned “navigation via triplets”. This is not the place to explain how this algorithm functions. The details are programming oriented, as a “data sample” to aid debugging.

Interleaving



Interleaving combines two streams: Blocks of 456 bits that open in bursts of 114 bits each, being the ratio 2:8, namely 2x456 bits open in 8X114.

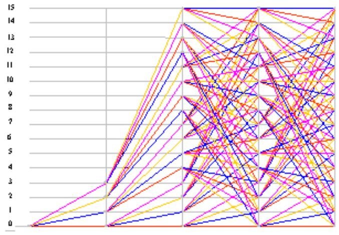
A central Block and its related tail half blocks provide bits following the rule depicted in the figure. B4/7:B5/3 means that burst "three" from left is built "interleaving" 57 bits from the seventh subblock of Block 4 with the 57 from the third subblock of Block 5. Bits are assembled from left to right, one for each subblock.

First bit always comes from the first member of the argument (B4/7:B5/3), in this case from B4/7.

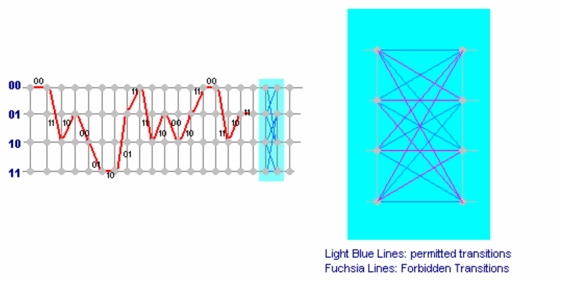
Warning!: observe that just in the middle of the interleaving process changes the ordering: B5/5 that follows B5/4 changes to be now in the left side of the relation now providing the leftmots bit.

Interleaving stands for a bits permutation within a given data block. Here is depicted the permutation to be performed via table lookups either to interleave or de-interleave.

Trellis - States

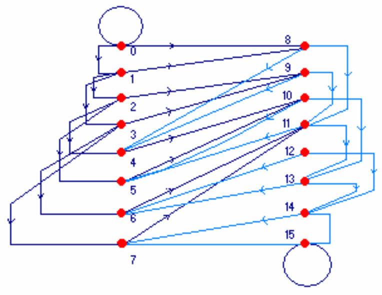


A classic Trellis algorithm display applied to a 16 states machine

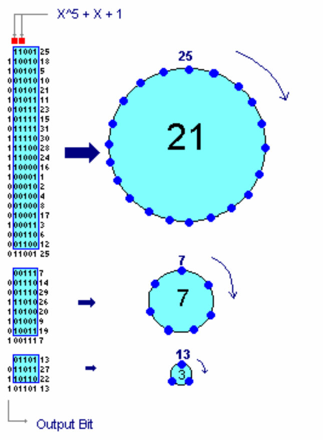


Light Blue Lines: permitted transitions
Fuchsia Lines: Forbidden Transitions

A 4 states machine Trellis going forwards step



A peculiar States Machine graph, used for example for Convolution coding.



```
  11001 25
1 10010 18
1 00101 5
0 01010 10
0 10101 21
1 01011 11
0 10111 23
1 01111 15
0 11111 31
1 11110 30
1 11100 28
1 11000 24
1 10000 16
1 00001 1
0 00010 2
0 00100 4
0 01000 8
0 10001 17
1 00011 3
0 00110 6
0 01100 12
0 11001 25

  00111 7
0 01110 14
0 01110 29
1 11010 26
1 10100 20
1 01001 9
0 10011 19
1 00111 7

  01101 13
0 11011 27
1 10110 22
1 01101 13
```

A data sample for a script that detects parasitic cycles within pseudo random virtual machines.