ChainFinder code for searching of chains

E.G.Romanov roeug@lycos.com

http://snow.prohosting.com/roeug/index.htm or http://www.websamba.com/roeug/default.htm **(May, 2003)**

The main purpose of program ChainFinder is to search for transmutation chains that could be realized for isotope production by means of irradiation in nuclear reactors. Though the isotope producing chains are mainly well known, the problem is to produce new nuclides for which ways of production are unknown. In addition to this, sometimes it is necessary to find out the reasons of some impurity nuclides presence in the irradiated material. After chemical analysis by which the element structure of a starting material and the constructional materials of the irradiated devices is completely determined it is possible to establish the isotope structure of the target. From the analysis of the ready product (gamma-, beta-, alpha- spectrometry) the set of radioactive isotopes that have been produced at the irradiation is defined. In ChainFinder program both the starting isotope and the final one are input and the code automatically (using ORIP_XXI software suite data file) tries to construct chains of nuclear transmutations (Fig.1). During the chain search process the data on radioactive decays branching are taken into account. For example, at Zr-95 beta- decay the daughter nuclide Nb-95 is formed in both the ground and metastable states. Thus, the data used are not only those displayed by NKE program but also others available in the software suite data file.

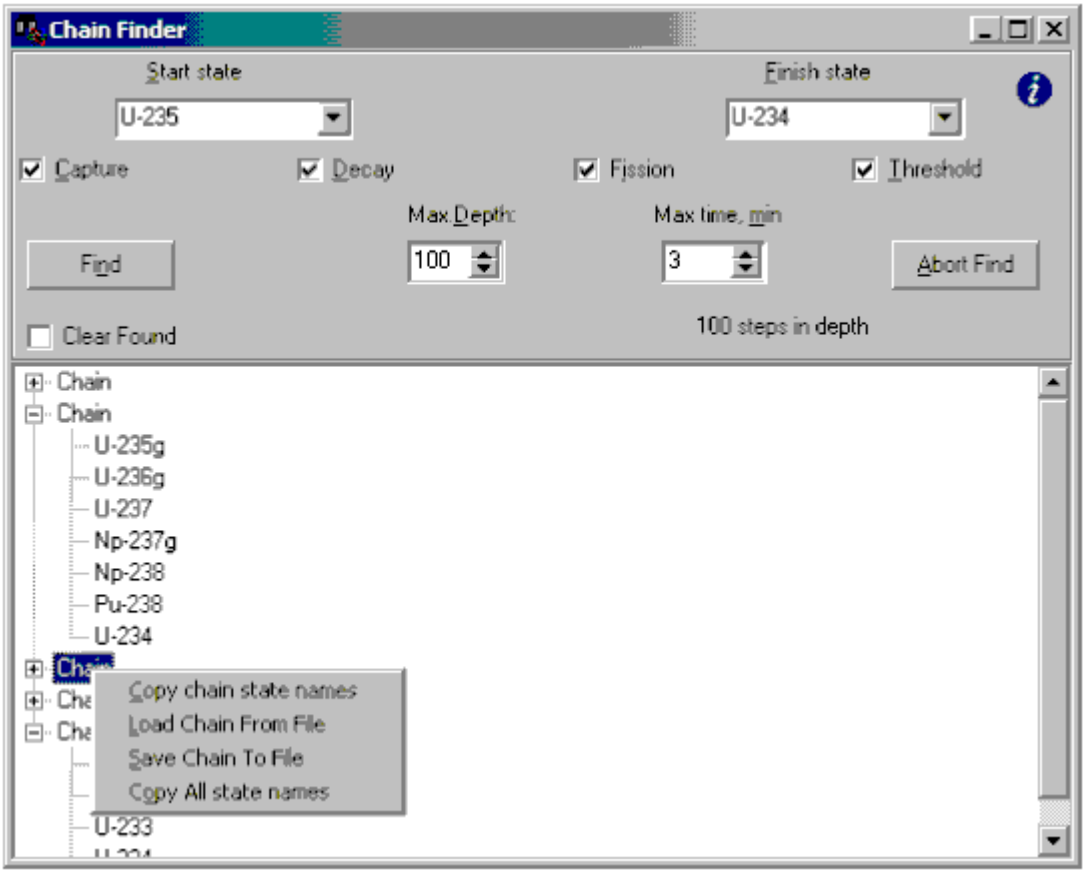


Fig.1. ChainFinder program main window

The pop up menu (right mouse button click) allows to copy the list of the chain state names into Windows clipboard or to save/load selected chain in/from an external file which could be used in ChainSolver program for transmutation calculations.

During a chain search it is possible to take into account or to omit reactions of radioactive decay, capture and fission (on thermal neutrons) and threshold reactions (due to fast neutrons) separately. For example, such specifications happen to be useful in the cases when it is necessary to carry out searches among transmutation chains at the storage and transportation time of non-fissionable materials when only decay chains exist. When fissionable heavy element isotopes are absent in a starting composition; disregard of fissions essentially reduces the search time. Since speeds of reactions on fast neutrons are usually low in a thermal neutron reactor; therefore, abandoned intent to consider threshold reactions will possibly make the analysis easier and faster.

Perhaps one of the most interesting examples for the program applications is the problem of U-234 production in nuclear reactor fuel (that is starting from U-235 and U-238 nuclei). Alongside with trivial transmutation due to fast neutrons Fig.2 other chains, more exotic ones Fig.3, are found. This fact allows to consider ChainFinder code as to be not very intellectual one but as an enough reliable program which will find set of unpromising chains of accumulation rather than miss any really important.

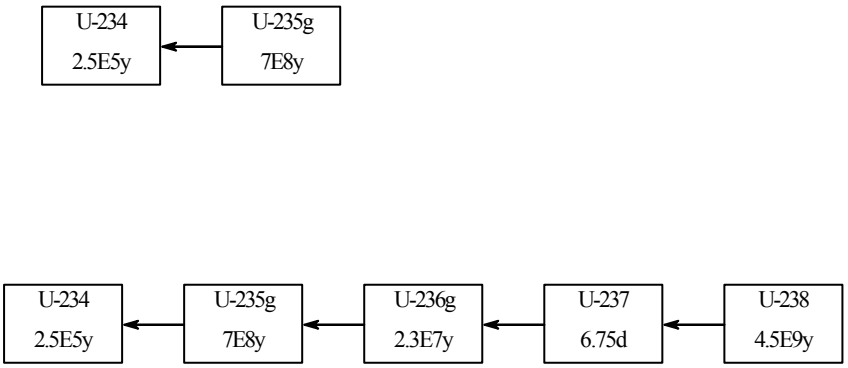
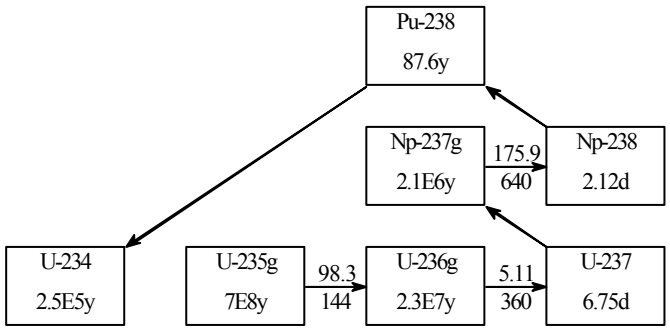
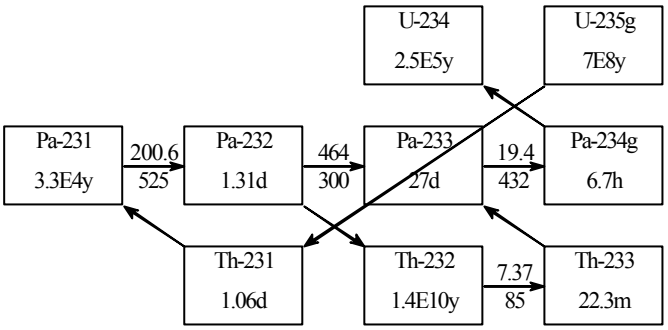
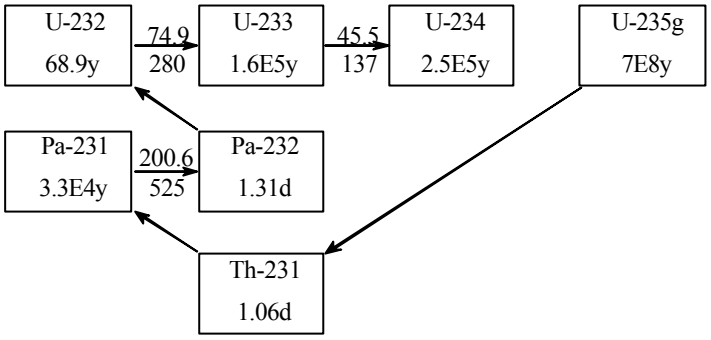
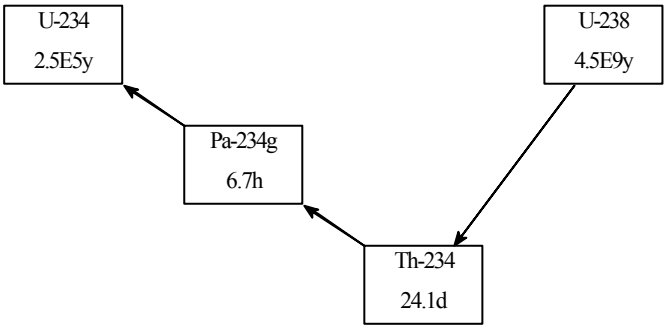


Fig.2 Trivial transmutation chains caused by (n, 2n) reactions due to fast neutrons




U-234
2.5E5y
U-235g
7E8y
Pa-231
3.3E4y
200.6
525
Pa-232
1.31d
464
300
Pa-233
27d
19.4
432
Pa-234g
6.7h
Th-231
1.06d
Th-232
1.4E10y
7.37
85
Th-233
22.3m


U-232
68.9y
74.9
280
U-233
1.6E5y
45.5
137
U-234
2.5E5y
U-235g
7E8y
Pa-231
3.3E4y
200.6
525
Pa-232
1.31d
Th-231
1.06d


U-234
2.5E5y
U-238
4.5E9y
Pa-234g
6.7h
Th-234
24.1d

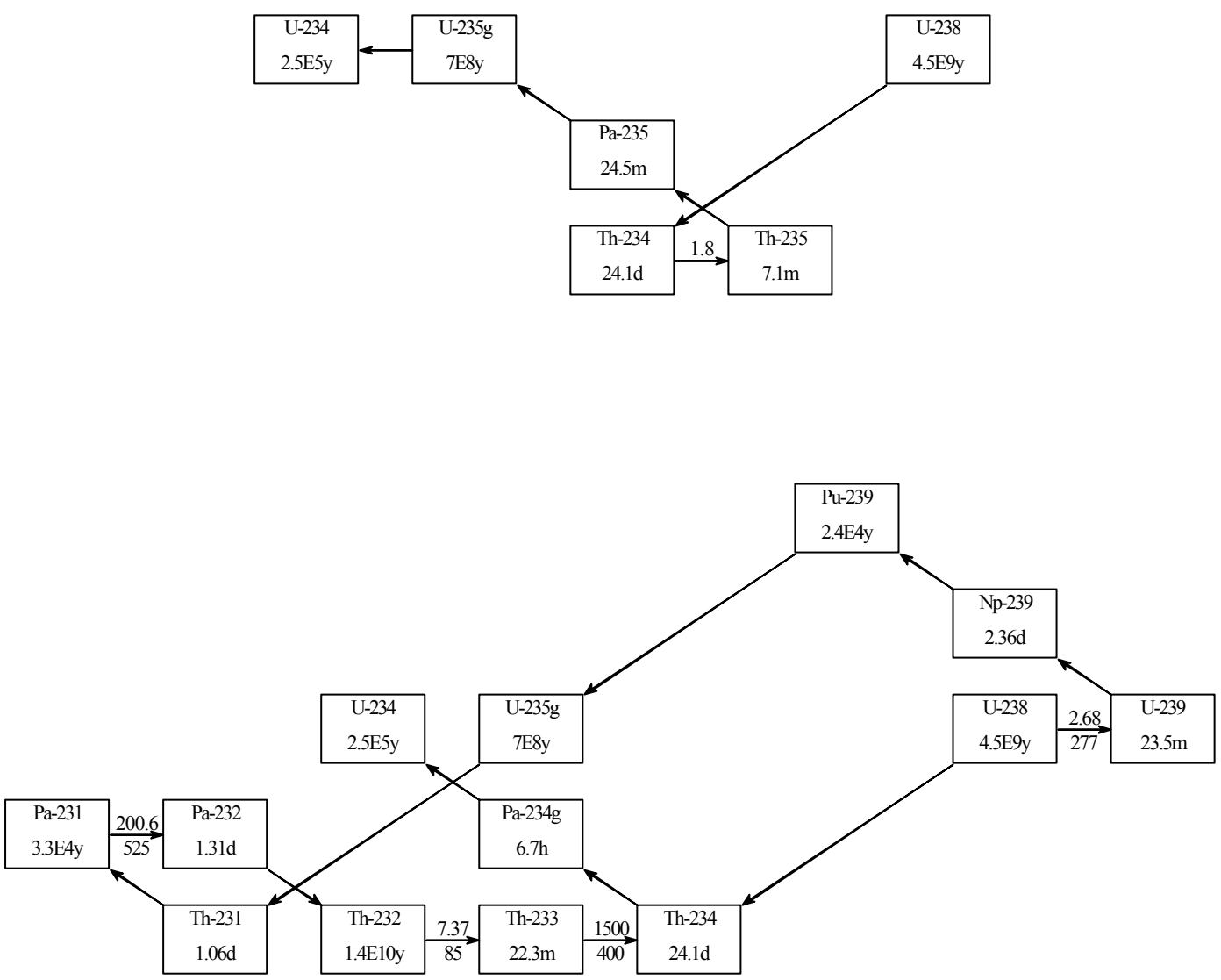


Fig.3. Complex transmutation chains

The algorithm of a transmutation chain search realized in the code is rather simple. At the first step all possible daughter isotopes for the starting nuclide are found (from handling the information in the software suite data file). On the following steps the starting nuclide is replaced with the found earlier isotope and the search subroutine is reintroduced. If the target nuclide or an earlier found isotope among daughter nuclides is present then the further search in this branch stops. If the target nuclide is found then the whole chain is constructed. Since, in any case, search on the only branch stops: therefore, the given algorithm allows finding a lot of (theoretically all) chains. The similar algorithm is realized in ChainFinder code and for a reverse search direction (from the target isotope to the starting one) then instead of daughter states possible parents appear. Similar algorithms of path finding now are widely used in programming systems of computer aided design (trace wave algorithms in systems of designing printed circuit boards, for example) and game programming ("search in width" and "search in depth", [1]).

Generally speaking, the theory and practice of programming of path finding on the complex surface is constantly arising in game programming. These problems are apparently closely to ones for ChainFinder program. Some algorithms of a path finding are known: Dijkstra algorithm, algorithm "the best is the first", * algorithm ("A-star") [2]. But all of them are intended for paths on initially known (or practically completely known) map and thus they are not applicable to a problem of nuclear transformation chain search. One of the contemporary algorithms of path finding D* ("D-star", [3]) works in those situations when the information about outskirts is known only at the movement before the next step to be done. This situation is very close to ChainFinder problems. The given algorithm allows finding in the shortest time a path with the least "cost" (the sum of "expenses" for separate "steps"). In this case it is necessary to define the function of expenses for each production of a next nuclide. At present, such function is not known for transmutation problems, mainly because of the fact that irradiation conditions can vary drastically during an irradiation. For example, a time stop of a reactor for a fuel reloading and neutron de-blocking because of blocked nuclides burning out give essential influence of neutron reactions velocities, hence, on transmutation "costs". Apparently, the characteristic appropriate to the cost may be revealed and chosen only in practical operations of ORIP_XXI software suite, and after use of ChainFider together with ChainSolver codes. Besides, algorithm D* is to find unique (though the best) path (in our case a chain of isotope production) while ChainFinder code tries to find out all possible chains to be realized during the irradiation.

ChainFinder program tries to carry out the preliminary analysis of branching chains to choose essentially distinguished branches. For the explanation of the above statement it is possible to consider the following example. Let us suppose that some intermediate nuclide may be produced from a starting isotope by three different chains. Let there exist three different ways to produce the target from this intermediate nuclide. Then there are nine (3*3) chains of transmutations from which only six should be displayed: the three former chains from the starting isotope up to the intermediate one and three more chains from the intermediate nuclide up to the target.

Though ChainFider code is a part of ORIP_XXI software suite the program may be used as the stand alone utility (together with the data file).

References

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, Introduction to Algorithms. The MIT Press/McGraw-Hill, 1990

[2] N J Nilsson, Principles of Artificial Intelligence, Springer Verlag. Berlin, 1982

[3] Anthony Stentz, Optimal and Efficient Path Planning for Partially-Known Environments, In Proceedings IEEE International Conference on Robotics and Automation, May 1994