

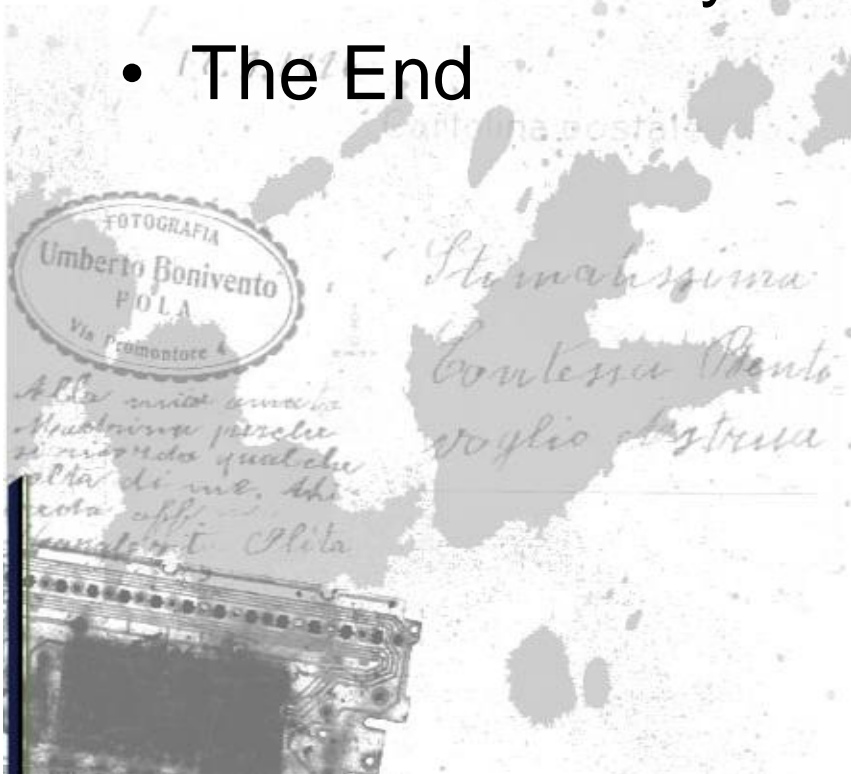


Static Detection of Vulnerability by Data Flow Analysis

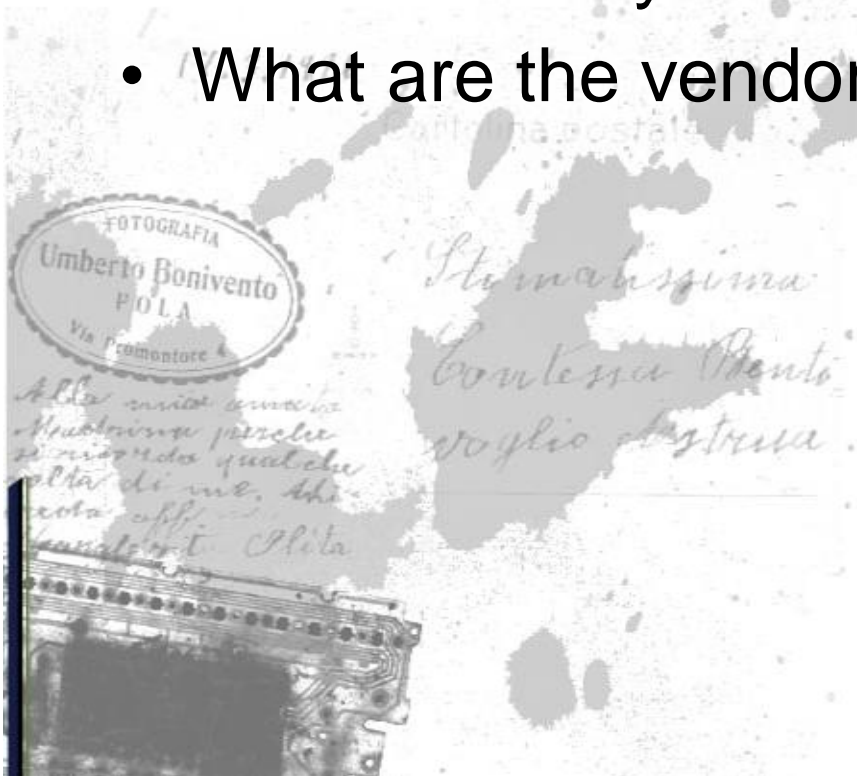
funnywei@xfocus.org



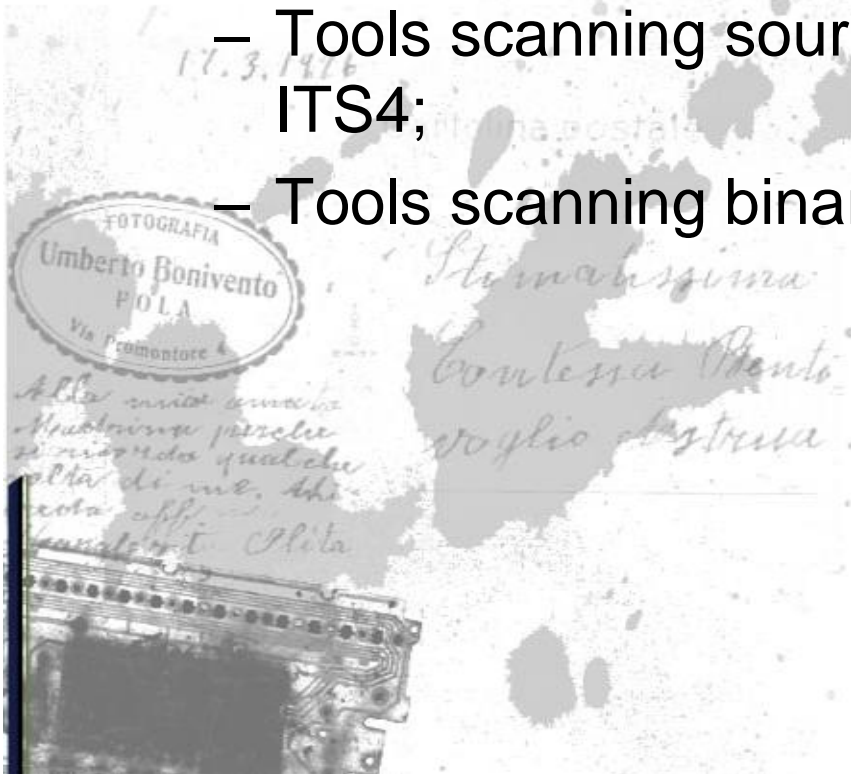
- Review
- System Architecture
- Intermedia Representation Language
- Data Flow Analysis
- The End



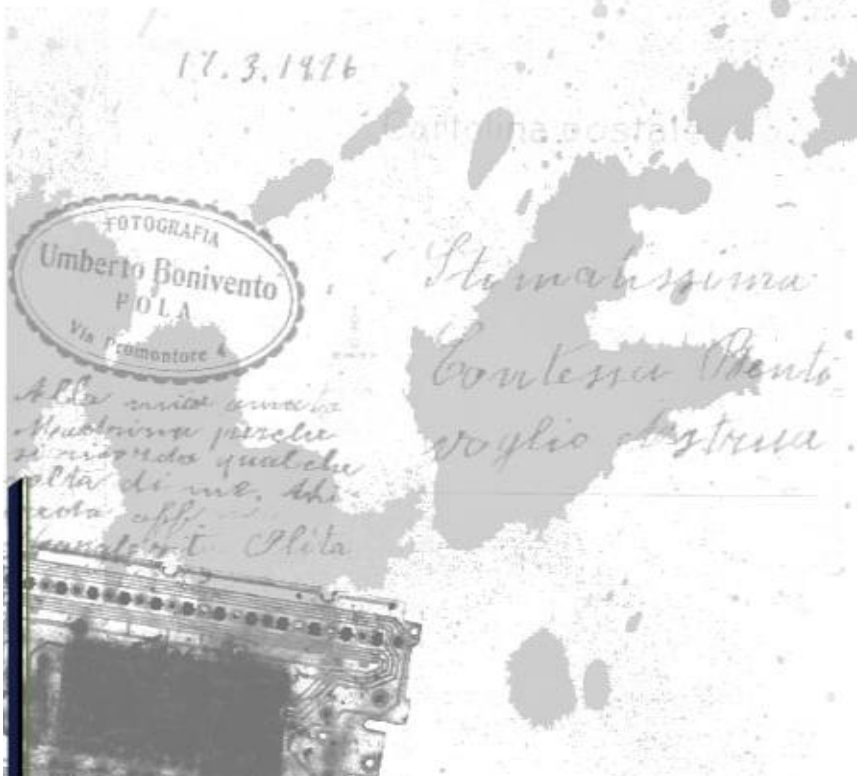
- The early automated scripts, tools
- The Advantages of these tools
- Existed problems
- Our deficiency
- What are the vendors doing?



- Basic ideas:
 - Pattern matching;
 - Limited back trace.
- Examples
 - Tools scanning source codes, such as FlowFinder, ITS4;
 - Tools scanning binary objects, series of idc scripts.



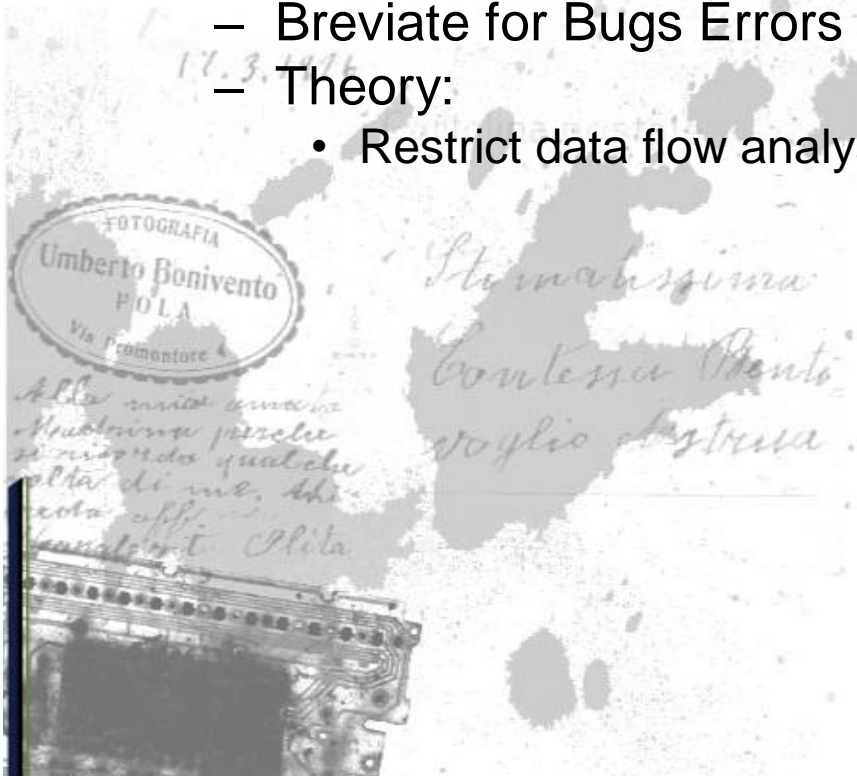
- Easy to develop;
- Run faster;
- Lower resource consuming;

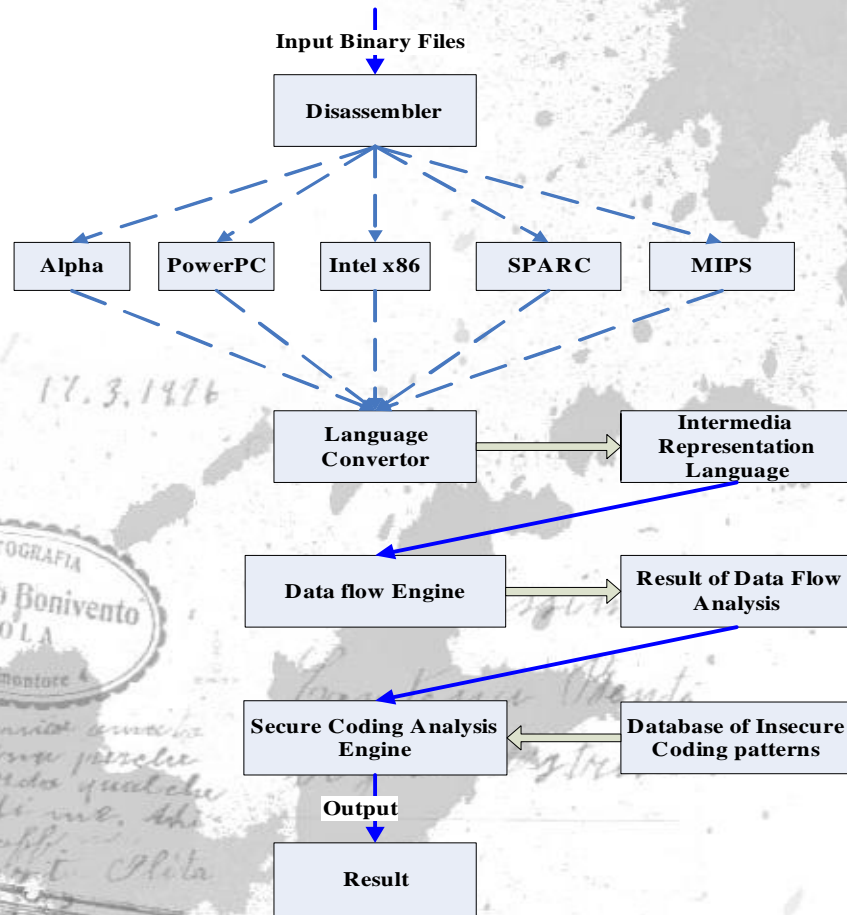


- Cpu dependence;
- Too much false negatives, for the reason of the complexity of the problems coming from the real world;
- false positives! Our passions are flooded in the valueless points;
- Can not understand the program's semantics, and only analysis the intra-procedure part.

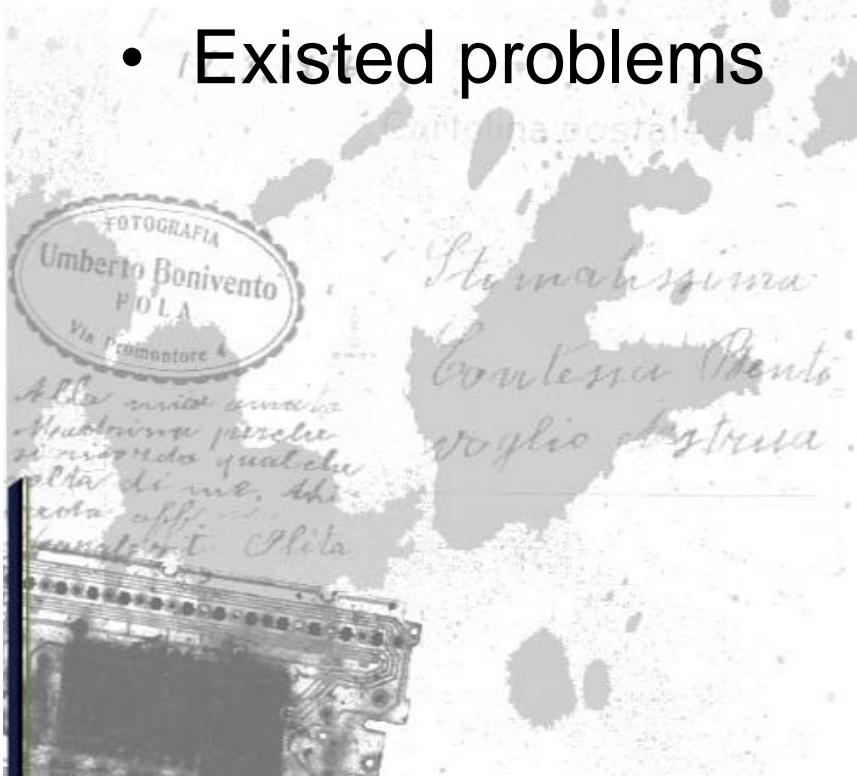
- Pure experimentalism!
- lack of theory's support, don't know how to do it better.
- The period of the analysis is too long, and we need more automated tools.
- The tools need understand the programs.

- Microsoft's SLAM projects
 - 1999, 60M \$, purchase intrinsic's prefix to control win2k's bugs.
 - Develops SLAM projects, detect bugs in drivers.
 - Theory:
 - Program Analysis, Model Checking, Automated Deduction
- IBM BEAM
 - Breviate for Bugs Errors And Mistakes
 - Theory:
 - Restrict data flow analysis to path that are executable.

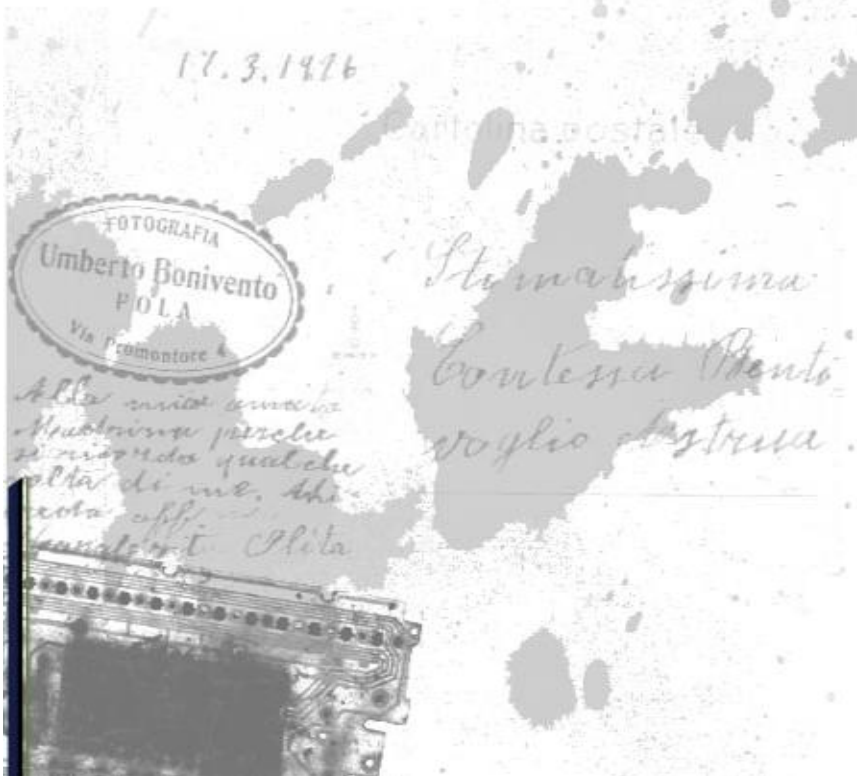




- How to design?
- The form of intermedia Representation Language
- Our designing
- Existed problems



- Steven S. Muchnick, Intermediate-language design is largely an art, not a science.
- Refer to other's fruits
 - Halvar Flake's MetaCPU



- Existed Intermedia Representation
 - Polish prefix, quadruple, triple
- Quadruple is three address representation. The form is:
 - (op, arg1, arg2, result)
- When op has one or zero arg:
 - (op, arg1, ---, result) or (op, ---, ---, result)
- Triple is another form of quadruple, for reduce the temporary variables.

- Refer to the Halvar Flake's design
 - Risc-like
 - Sparc-like
 - No limitation of the num of the registers
 - 256 global regs, %g00-%gFF
 - 256 temporary regs,%t00-%tFF
 - 256 I/O regs, %i00-%iFF/%o00-%oFF
 - 256 flag regs, %f00-%fFF
 - PC, SP, FP

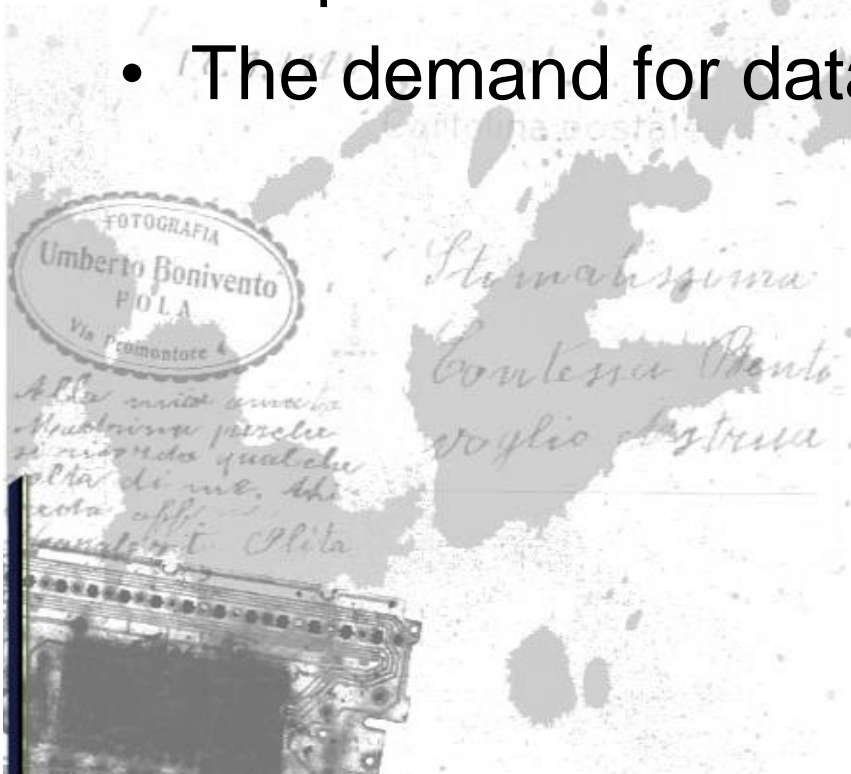
- Explicit accessing memory instructions
 - Mem as source address
 - mov reg, mem
 - ldm mem, ---, reg
 - Mem as the destination address
 - mov mem, reg
 - stm reg, ---, mem
- Explicit loop copy instructions
 - 17-3. rep movsd
 - cmp gXX, 0
 - br_nz XXXXXXXXX
 - dec gXX, 000004, gXX
 - ldm/stm
 - loop XXXXXXXXX

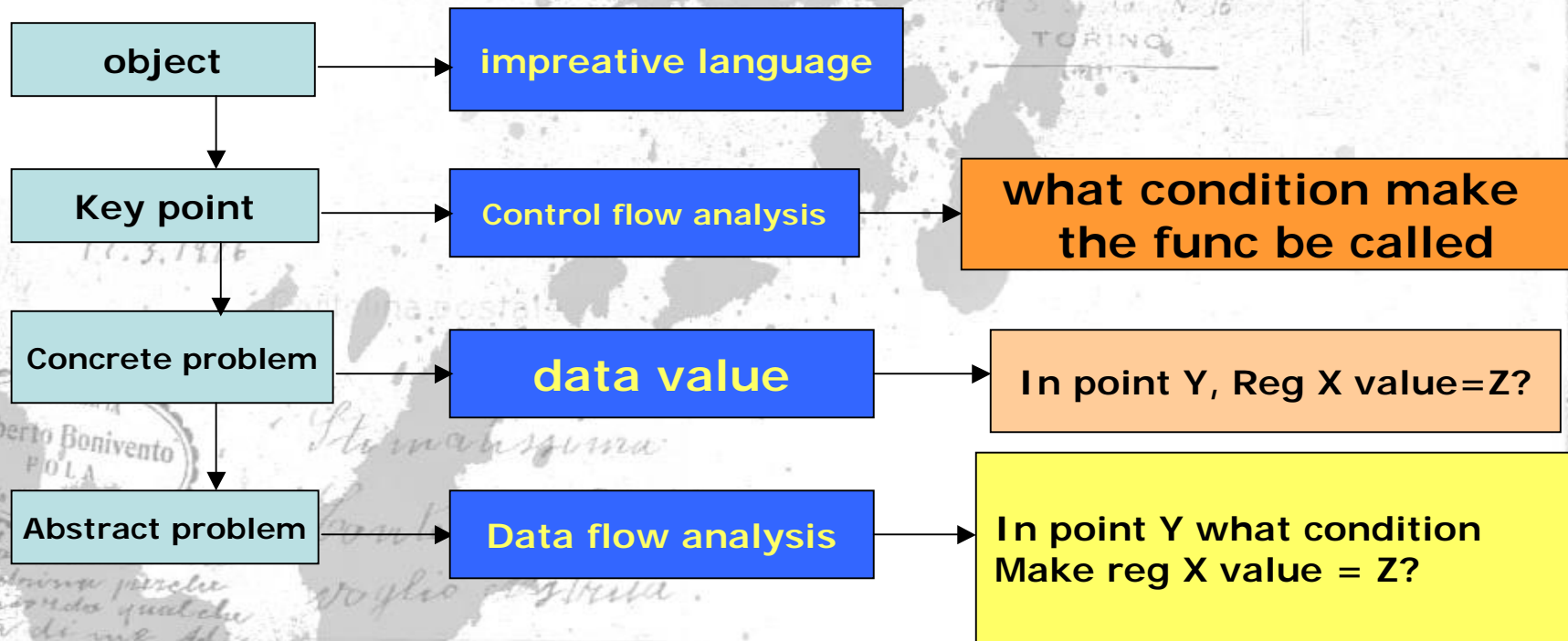
- Theory:
 - Can not prove the correctness of the translation;
 - Context-free grammar recognition → Context-sensitive grammar recognition.
- Practice:
 - Compiler optimization, need to be recognized.
 - The different optimization of different compilers will be translation's trouble maker!
 - For example, mov replace push, we must translate mov [esp+offset] to the proper output reg.

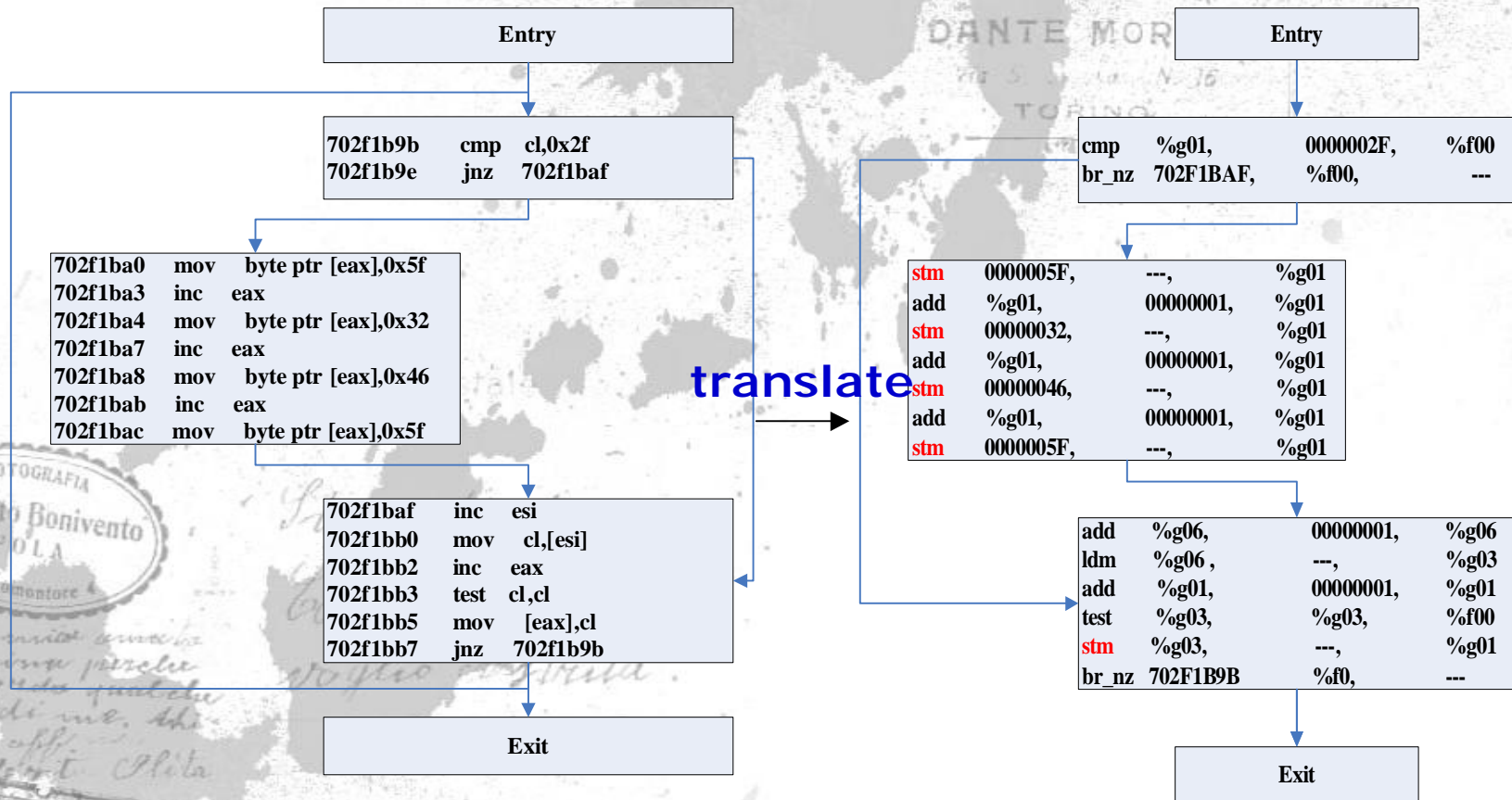
Before: push edx
Push ecx
call function
add esp, 8

After: sub esp, 8
mov [esp+0], ecx
mov [esp+4], eax
call function

- why use data flow analysis?
- IE Object Type Property overflow analysis
- Basic block analysis
- Loop detection
- The demand for data flow analysis





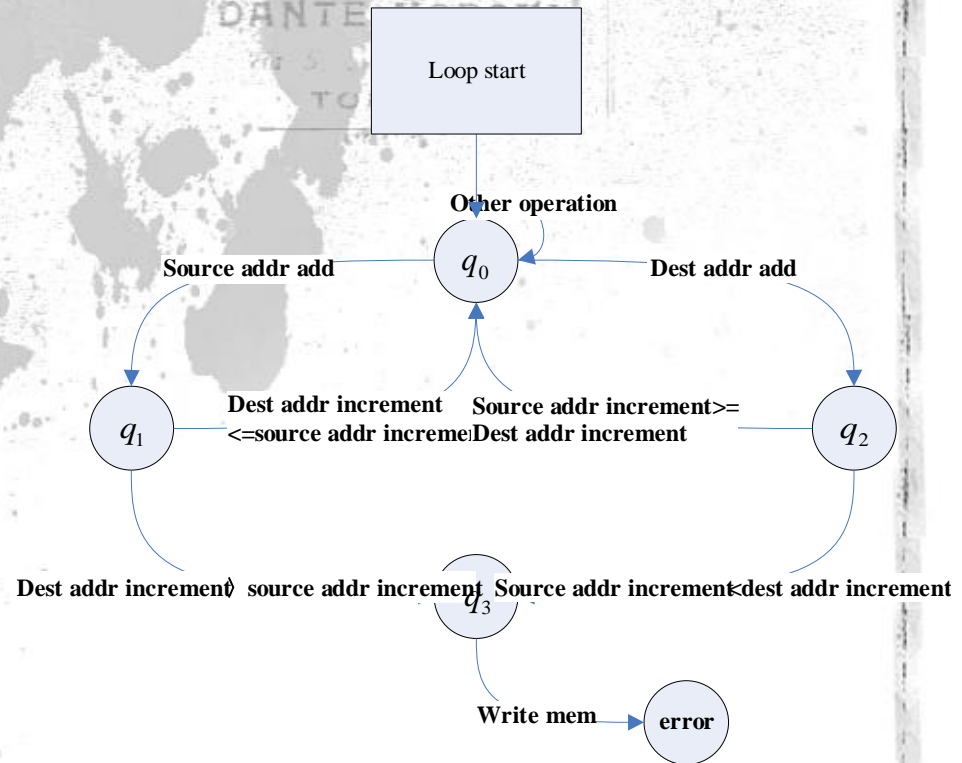


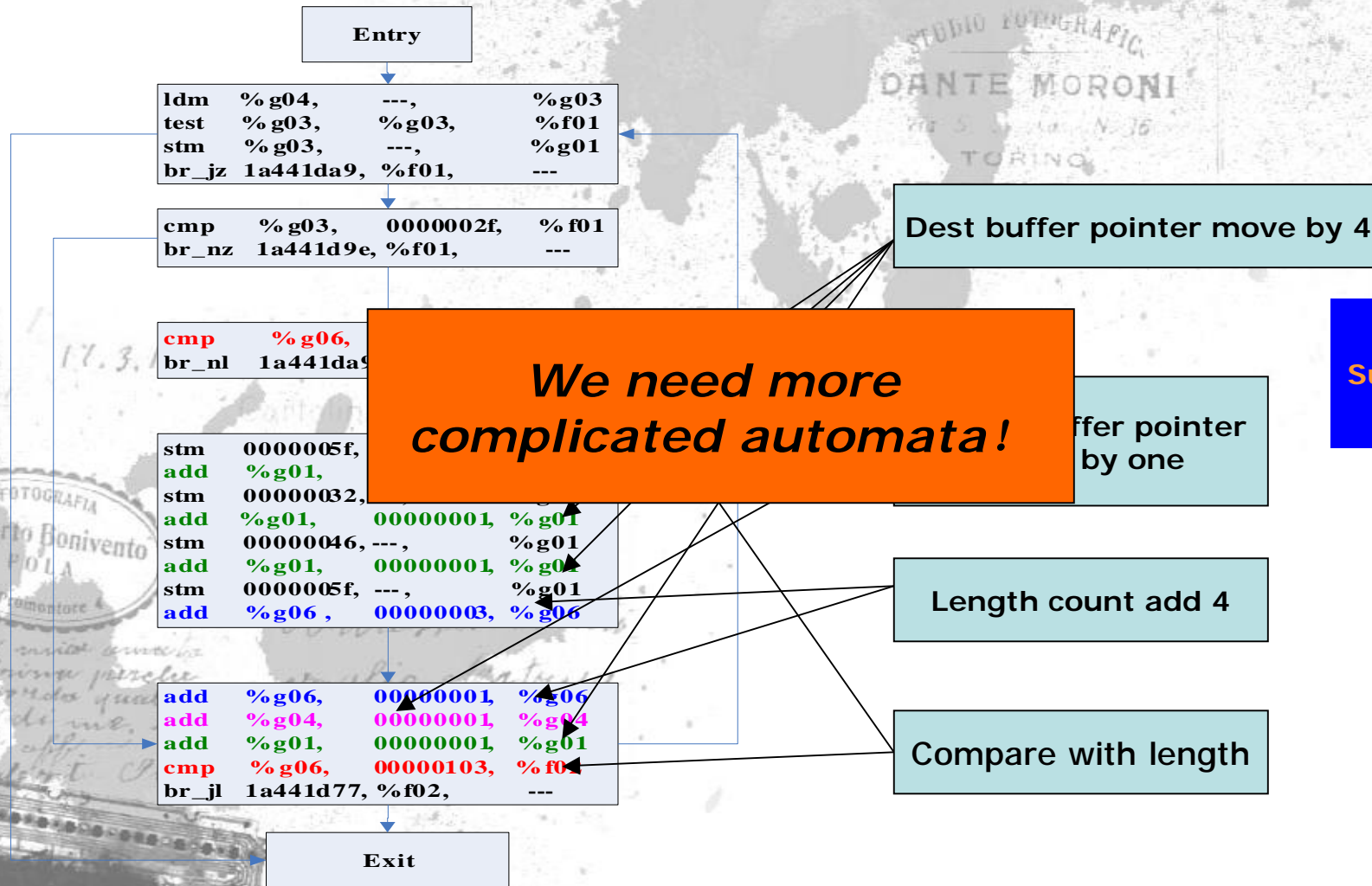
- If no stm in loops, not interested!
 - If each time write the same position, not interested!
 - If write a fixed number of bytes, not interested!
- This examples is very suspicious!

- We will use simplified automata to solve this sort of problems.



- PDA is defined to be a sextuple machine
- Initial states: q_0
- Final states: q_0 , error
- The execution can be described as the set of transition relation.
- Design more complicated automata
 - State transition for length comparison.
 - State transition for writing mem at a non-fixed point.





- Why not FSA
 - FSA is equal to type 3 grammar, but regular grammar can not describe the problem.
 - We need push symbols into stack!
- Why not Turing Machine
 - TM equal to type 0 grammar, too strong.
 - PDA equal to type 2 grammar, context-free grammar, enough!
- PDA classification
 - Terminates with final state (this example uses);
 - Terminates with empty stack.
 - Equivalence of both.

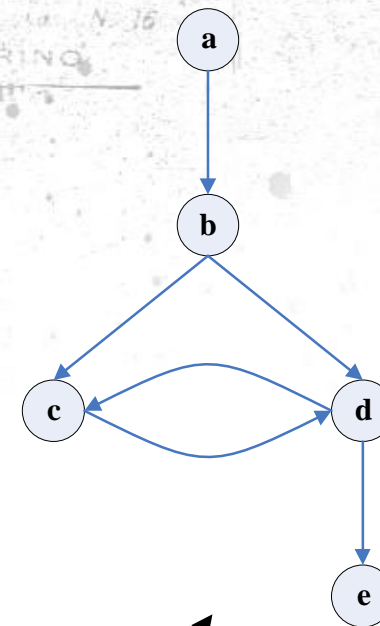
- Informally, A straight-line sequence of code that can be entered only at the beginning and exited only at the end
- The first instruction in a basic block may be
 - The entry point of the routine,
 - A target of a branch, or
 - An instruction immediately following a branch or a return.



- Determine the entry of each basic block.
- Through each entry, determine its including instruction. or when reaching the exit, all the instruction belong to the corresponding block.
- After the two steps, if one instruction is not included in any block, it is unreachable, we can delete it.

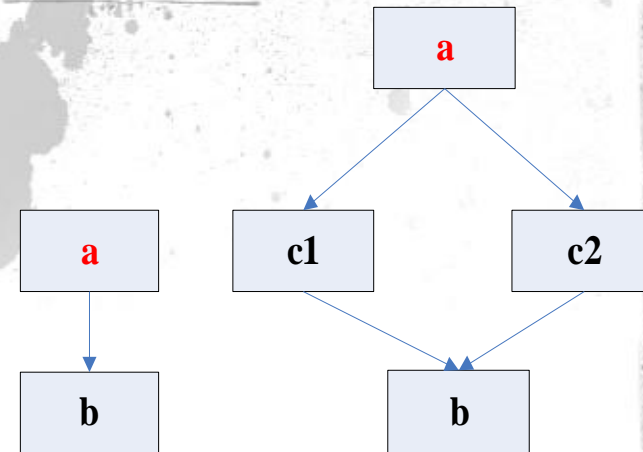


- What is loop?
 - Have a unique entry node, all the path from outside to inside, must pass the entry.
 - Nodes in this loops is strongly connected. Strong connection means that starting from one node, we can reach any other node (especially, when the set of nodes contains one node, it must have the directed edge to itself).

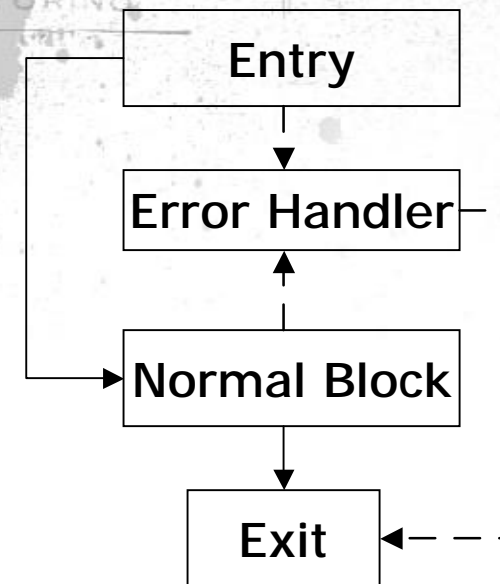


Not loop

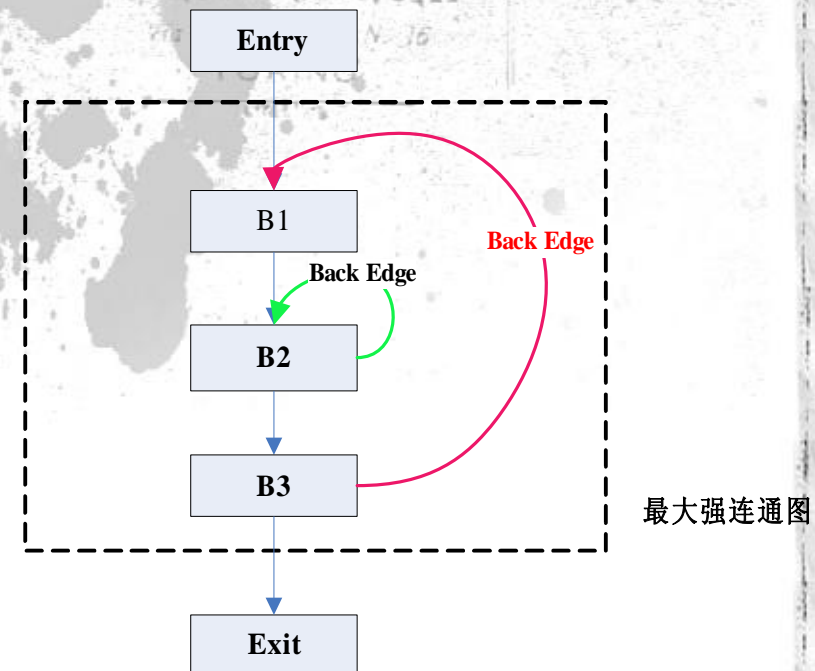
- Definition
 - We say that b dominates a if every possible execution path from entry to a includes b .
- Two approaches to compute the set of the dominators:
 - a dominates b if and only if $a=b$, or a is the unique immediate predecessor of b , or b has more than one immediate predecessor and for all immediate predecessors c of b , $c \neq a$ and a dominates c .
 - Lengauer and Tarjan's algorithm, faster but more complicated.



- C Language's Exception Mechanism
 - The call to setjmp will become the header of a new block.
 - Windows exception handler instruction



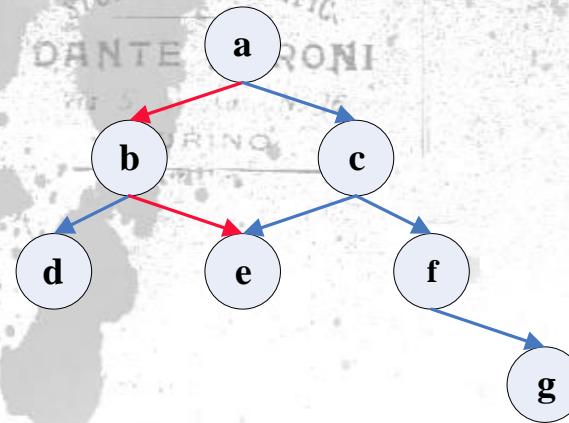
- Back edge in a flowgraph as one whose head dominates its tail.
- Given a back edge $n \rightarrow d$, the loop of $n \rightarrow d$ is the subgraph consisting of the set of nodes containing d and all the nodes from which n can be reached in the flowgraph without passing through d and the edge set connecting all the nodes in its node set. Node d is the unique entry.



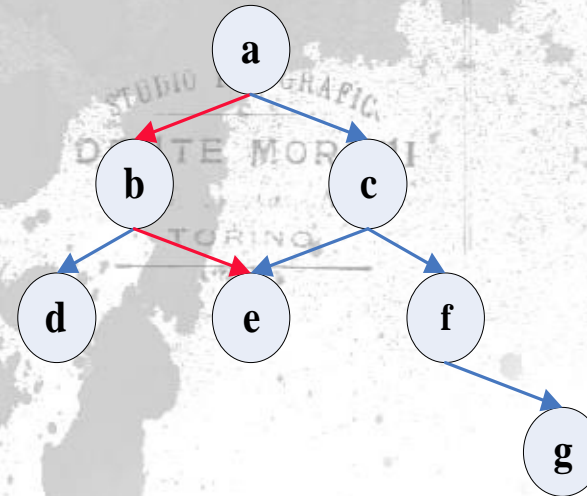
- Intra-procedure analysis
 - Locate the suspicious spots
 - Contain:
 - Basic Block analysis
 - U-D chain
 - Structure analysis
- inter-procedure analysis
 - Vuls always produced by several functions interaction.
 - The formation process of the buffer some times is not easy to judge.

► Use the both to get better effect.

- Variable's value?
- If function e is vulnerable.
- When error occurs, the exec path is
 - a->b->e
- We often meet the question?
 - When will function a branch b?
 - In point Y, when the value in reg X will be equal to Z?

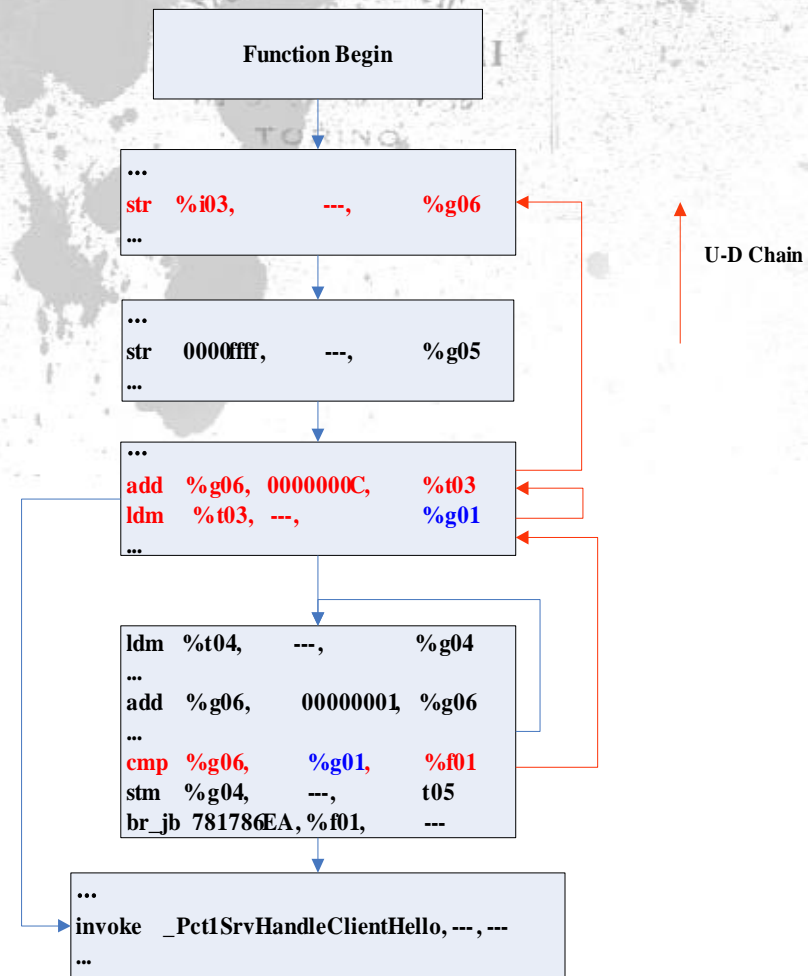


- The construction of the data
- If bof may occur in e, we should check:
 - length of the buffer in b have been defined or compared for limitation?
 - Or comparison occurs in a?
 - The construction of buffer is very complicated, a, b and c take part in the data building.



Combine the Intra and Inter-Procedure analysis

- Combine the U-D chain analysis and I/O registers usage
- The complete data flow analysis must be the NP problem
- Use U-D chain analysis IIS PCT Vulnerability



- Formal Language
- Formal Semantic, especially operation semantic
- Graph Theory
- Discrete Mathematics
- Type Theory
- Compiler design and implementation
- Program analysis

- Thank u!
- My email
 - funnywei@xfocus.org
 - funnywei@163.com

