On Program Plan Recognition in Legacy Code

Steven Woods and Qiang Yang

Department of Computer Science

University of Waterloo,

Waterloo, Ontario, CANADA

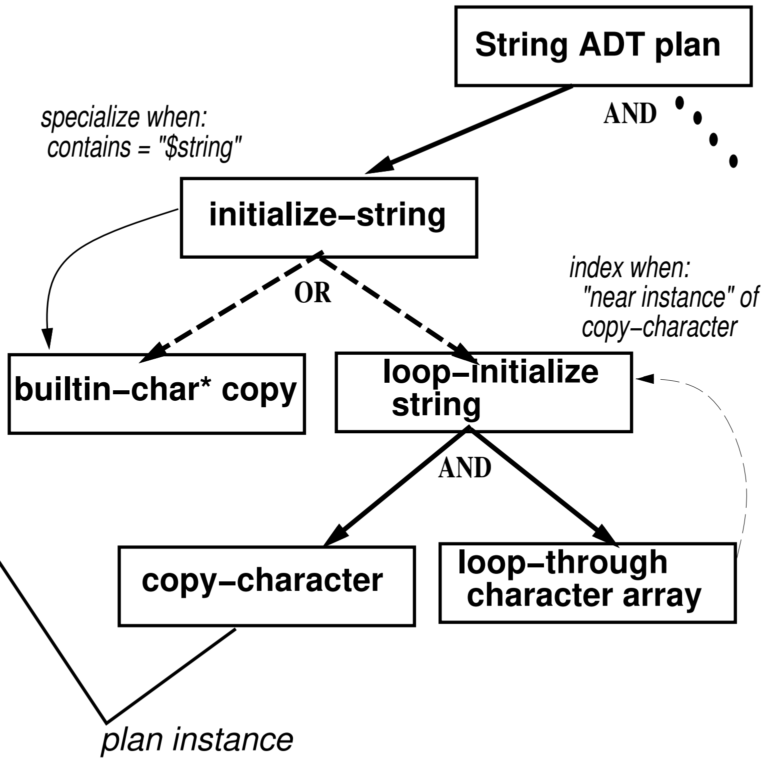
August, 1995

Understanding Legacy Code : Knowledge Mapping

Legacy Source Code

```
main()
{
  char* A, B, C;
  ...
  A = "s" + "t" + "r" + "i" + "n" + "g" +  "1";
   ...
  B = "string 2";
  ...
  sz = 7;
  for (int j = sz; j > 0; j−−) {
      ....
      C[sz − j] = B[sz − j];
      ... }
  ...
  C[sz] = 3;
  ...
  for (int i=0; B[i]; i++)
      ...
     print(B[i])
      ....
  ...
  for (int j=0; C[j];j++) {
      ....
      printf("%s",C[i]);
      ... }
  ...
  for (int k=0;A[k]; k++) {
      ...
      outchar(A[k]);
      ... }
  ...}
```

Program Plan Library (excerpt)



→ **How may we automate part of this mapping process ?**

Approach : Constraint Satisfaction Paradigm

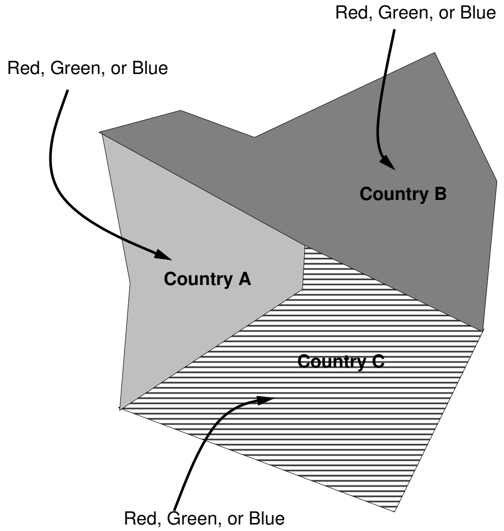


Figure 1: Map Color Example.

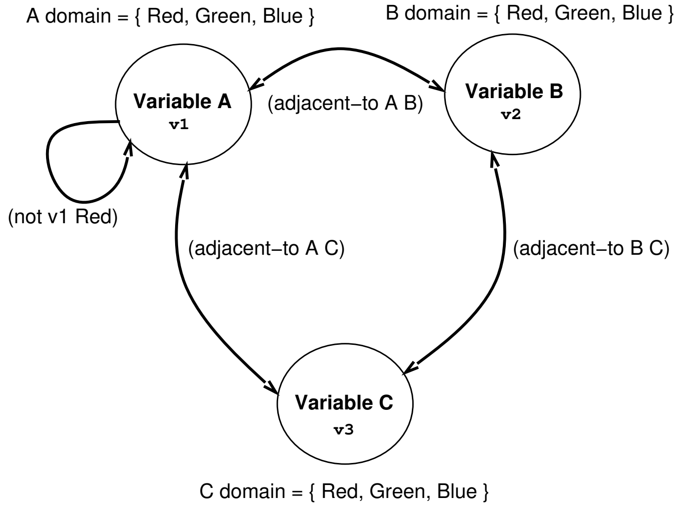
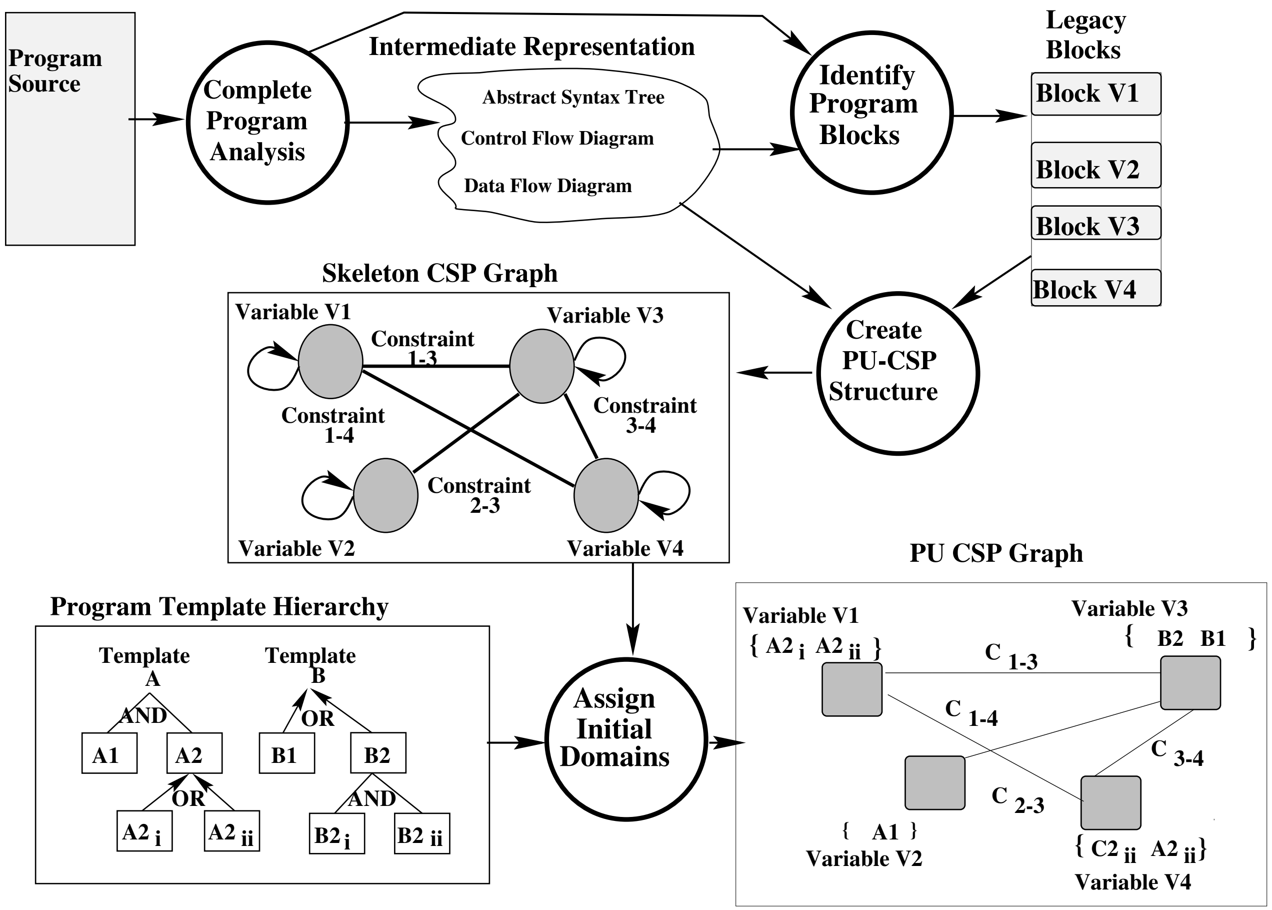


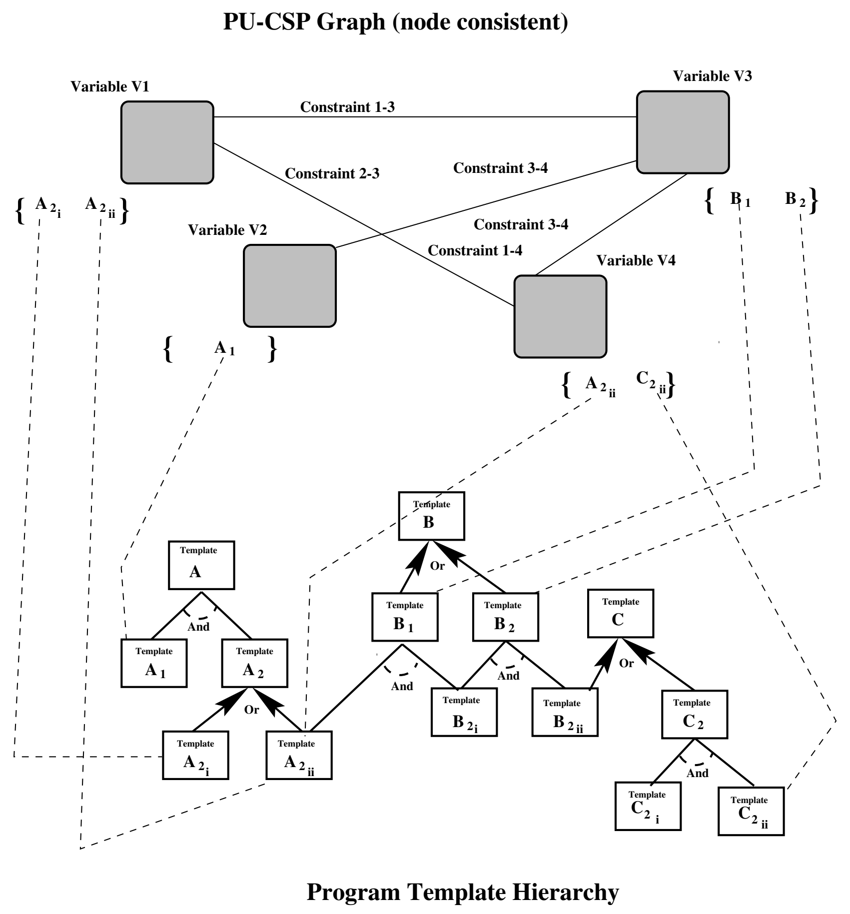
Figure 2: Map Color CSP.

- Known solution approaches (Search, Propagation, Hybrid)
- Well constrained problems $\rightarrow$ reasonable performance
- Domain knowledge (i.e. plan library) $\rightarrow$ tight constraints

Program Understanding CSP



Program Understanding CSP: A detailed example



Formal Characterization and Solution

- PU-CSP is NP-hard
  - → Transformation from Subgraph Isomorphism
  - → (formal proof in a submitted ICSE paper)
- A large body of solution techniques exists in CSP
  - → Local Constraint Propagation Methods:
  - → – AC-, Path-, N-consistency
  - → Global, Intelligent Backtrack-based Methods
  - → – Variable-, Value- and Constraint-Ordering
  - → Local Search, G-SAT, Constraint Logic Programming
- Good Average-case Results

MAP-CSP: Plan Template/Cliché Matching as CSP

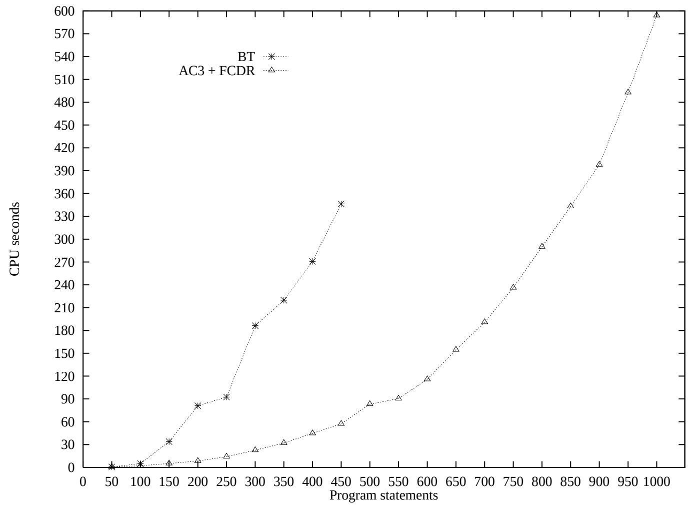
→ Finding all instances of a given plan template in a source code

→ The MAP-CSP problem is also NP-hard

→ Efficient CSP-based Algorithms Implemented.

→ Good Average-case Results

→ Test: Standard BT vs AC-3 with FC+DR (sec):



Related Efforts and Discussion

- Quilici (1994) Memory-based recognition, and Kozaczynski & Ning (1994)

  $\rightarrow$ Applies heuristic "tricks" to reduce search

- Wills (1992) Cliché concept recognition

  $\rightarrow$ Representation as clichés (grammars)

- Most can be unified under the CSP Framework

Ongoing Work

- Represent program plan *knowledge* & legacy source *structure* as constraints in **Understanding CSP**.
- Combine **Matching CSP** and **Understanding CSP** $\rightarrow$ integrated model of understanding including *Plan Under.* and *Rev. Eng.* heuristics
- Formulate partial library from industrial abstracted plans (BNR cooperation)
- Empirically demonstrate (partial) understanding of industrial source using **Understanding CSP**
- Identify knowledge and constraints req'd to give heuristic adequacy in interactive and batch instances