P7.Memory Pattern Analysis

Tiago Santos
CTM / HumanISE
INESC TEC

INESCTEC
TECNOLOGIA E CIÊNCIA
LABORATÓRIO ASSOCIADO
COORDENADO POR INESCPORTO

PEPCC

Detecting Patterns in DFGs

- Using graph isomorphism

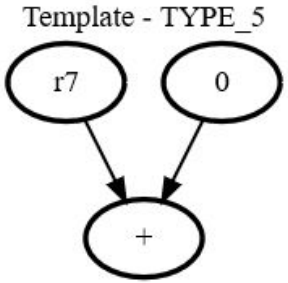
- Match to several patterns until one is found
  - Always start with the most complex templates first

- Matching based on the type of each vertex (load/store, arithmetic operation, immediate, register)
  - Can also consider the value itself, if necessary (e.g., vertices representing constants, such as 1 and 4, would match by type, but not by content)

Memory Patterns

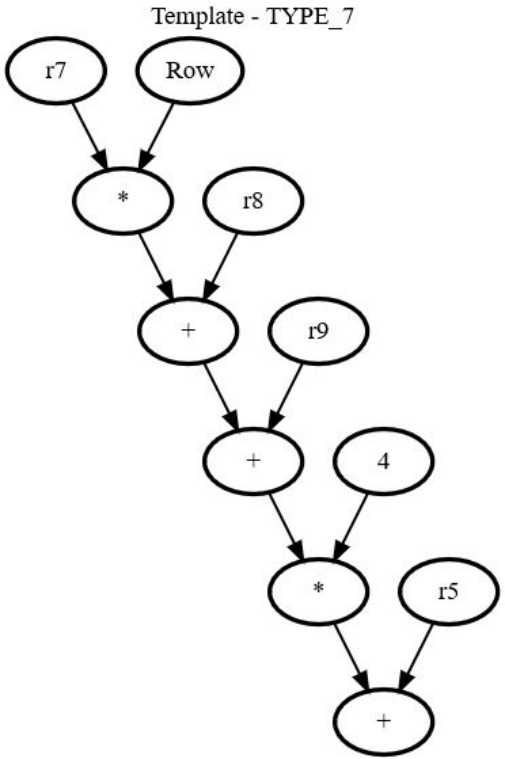
- 15 different patterns identified across PolyBench and Livermore

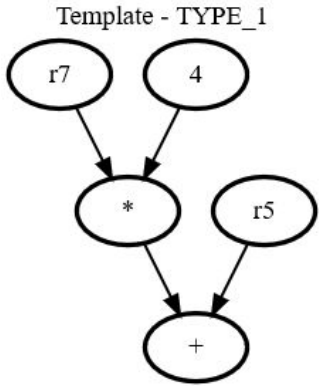
- DAGs with a single “add” operation as their sink
  - The same “add” operation implicit in a load/store instruction

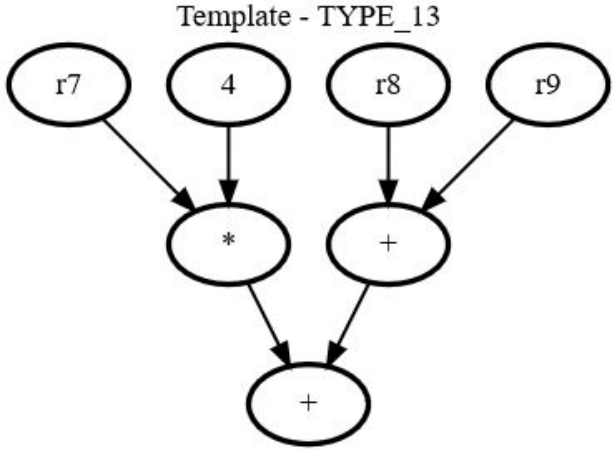


- Most basic template: Type 5 (address already in register + immediate offset)

Memory Patterns - examples

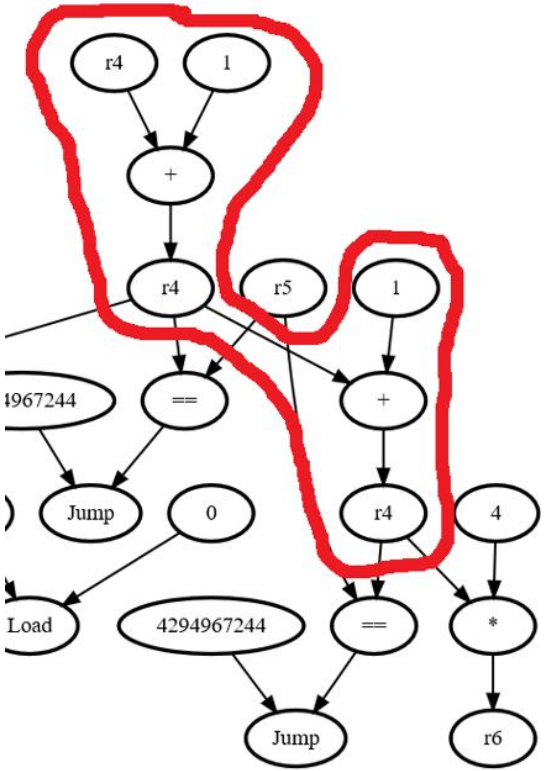






Identifying Streams

- For a memory access to be a stream, it needs to follow the properties:
- Incremented with a constant stride (verifiable by isomorphism)
- Memory expression is affine (determined by the pattern type)
- Should only take 2 iterations to verify if an access can be implemented as a stream



Replace accesses with Address Generation Units