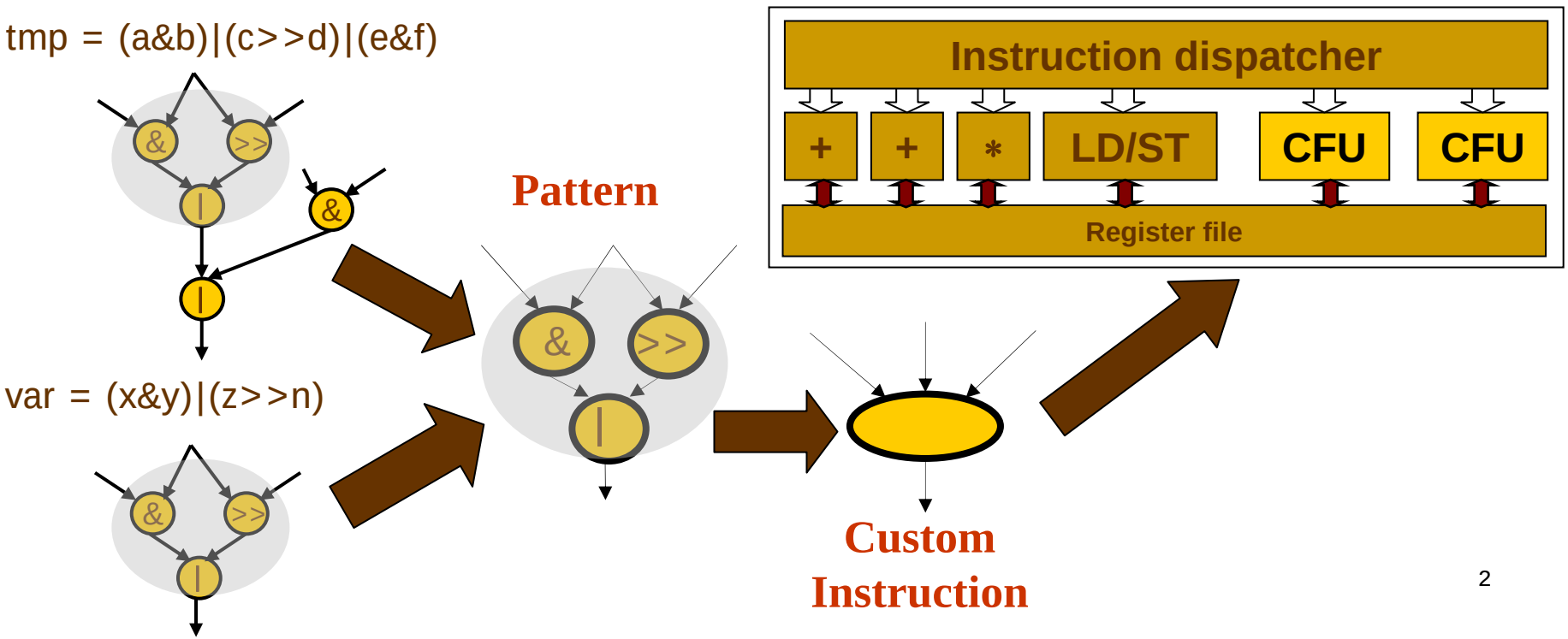
Satisfying Real-Time Constraints with Custom Instructions

Pan Yu, Tulika Mitra
School of Computing
National University of Singapore

Custom instructions

- Extend instruction-set architecture (ISA) of the processor core with application-specific instructions
- Micro-architectural support: Custom functional units (CFUs) to implement the custom instructions

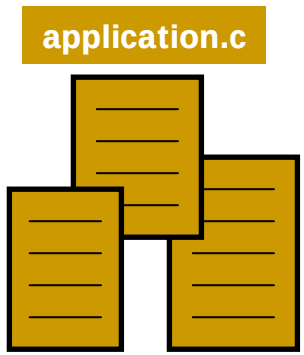


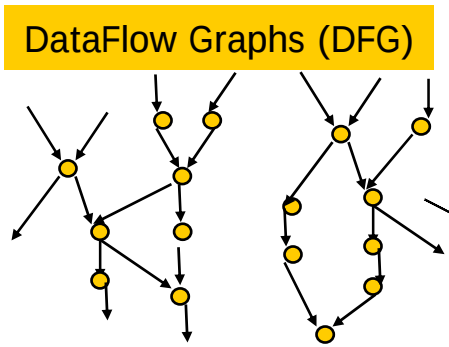
Custom instructions in real-time systems

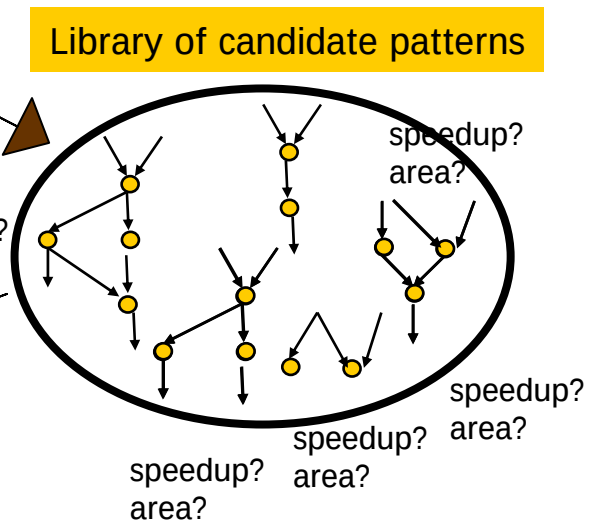
- Scenario: A set of optimized real-time tasks fails to meet deadlines
- Solutions:
  - Increase processor clock frequency
    - → Increased power consumption
  - Choose processor with higher performance
    - → May not be always feasible
- Can we exploit custom instructions to meet deadlines ?
- How do we select custom instructions for a real-time task?

Custom instructions selection

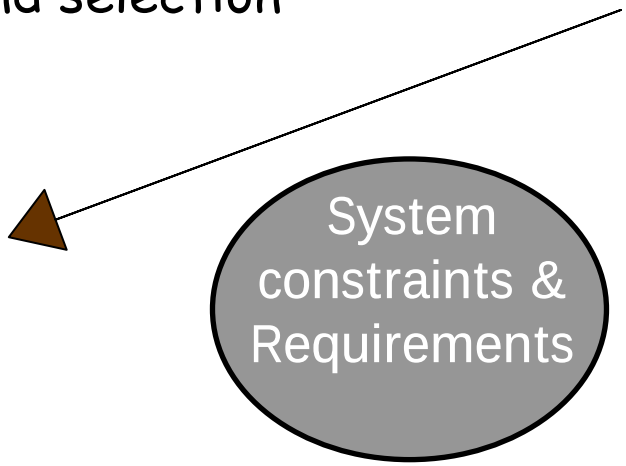
- Candidate patterns identification

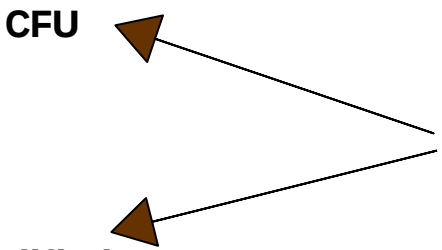






- Pattern evaluation and selection



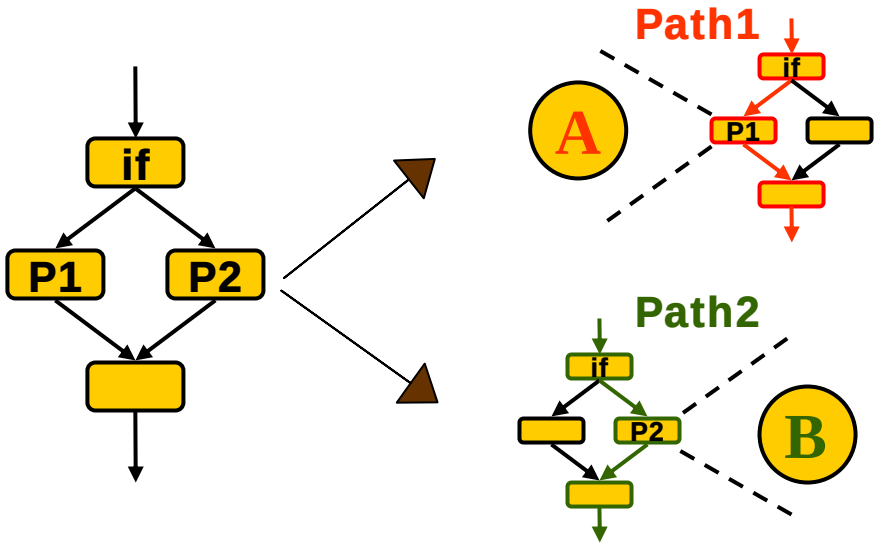


selected patterns

Do we need a new technique?

- Traditional custom instruction selection
  - Improves average-case execution time (ACET)
  - Relies on execution frequencies of patterns obtained through profiling
- Worst-case execution time (WCET) is the critical metric for real-time tasks
- WCET: Maximum execution time of a task for all possible inputs
  - Provides timing guarantees

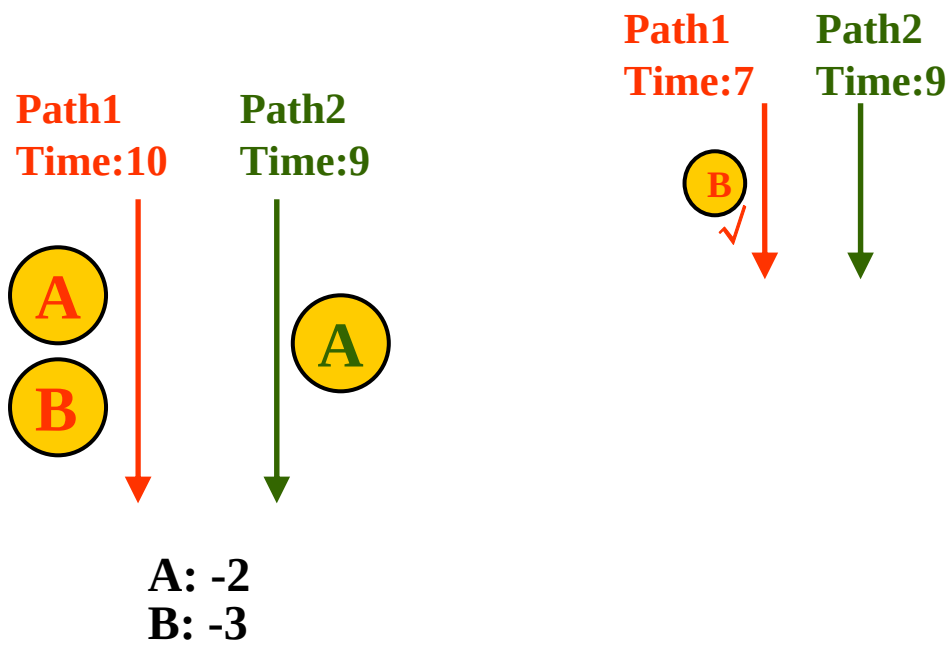
ACET *versus* WCET



- ACET-based selection chooses the pattern along the frequently executed path
- WCET-based selection chooses the pattern along the critical (WCET) path

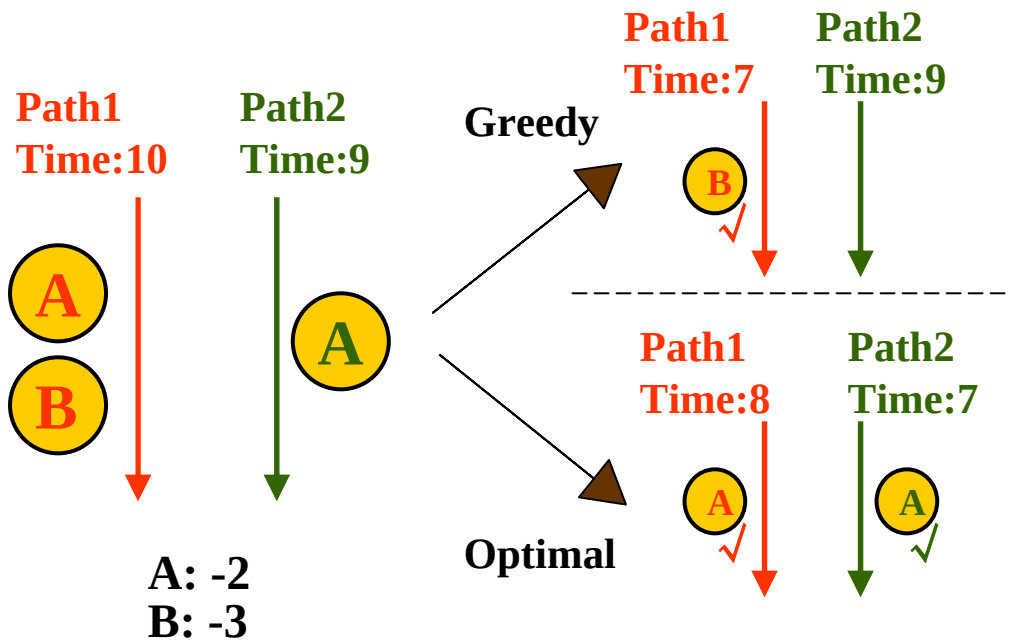
Challenges in reducing WCET

- Optimizing current WCET path may lead to a different WCET path



Challenges in reducing WCET

- Optimizing current WCET path may lead to a different WCET path
- Ignoring non-WCET paths may result in local optima



Proposed methodology

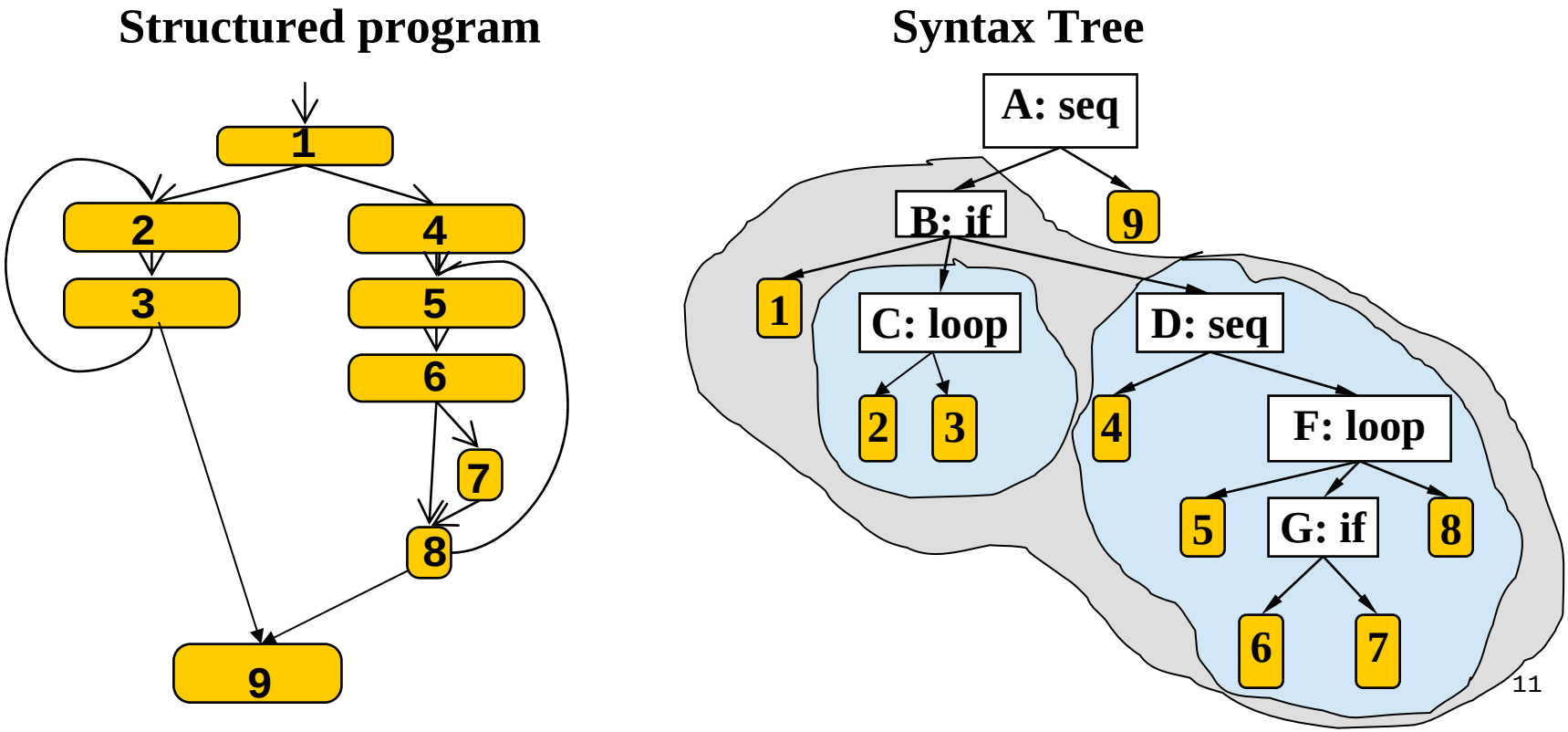
- Fast exhaustive enumeration of all possible patterns in the task [CASES'04]
- WCET-guided optimal or close to optimal selection of the patterns

Roadmap

- Introduction
- WCET Estimation: Timing schema
- Optimal solution
- Heuristic solution
- Experimental results
- Conclusion

WCET estimation: Timing schema

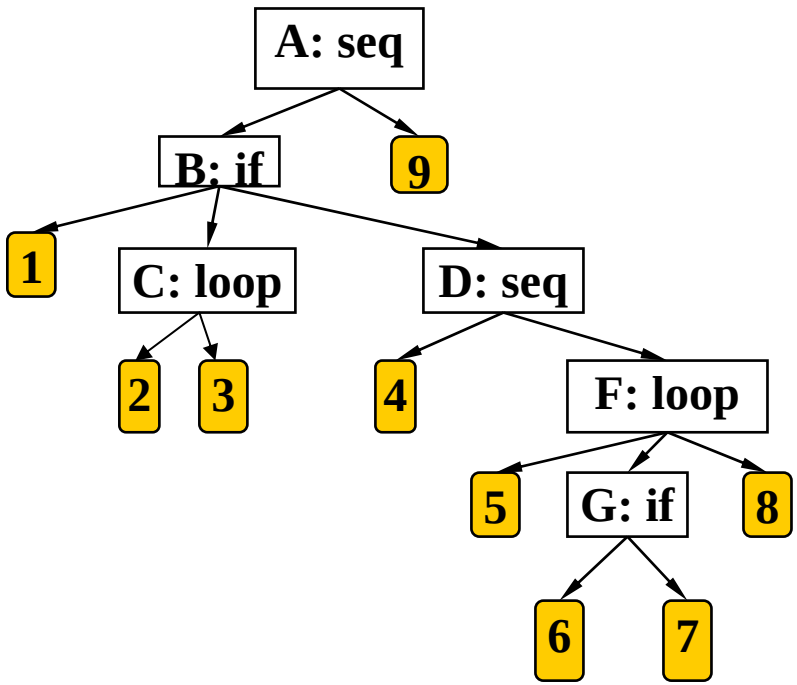
- Construct syntax tree with basic blocks as leaf nodes and control structures as interior nodes



WCET estimation: Timing schema

- Bottom-up traversal of syntax tree
- Apply timing rules at each interior node

Syntax Tree



Rules

Sequence: wcet(V1,…,Vn) =
wcet(V1) +… + wcet(Vn)

Branch: wcet(if V1 then V2 else V3) =
wcet(V1)+max(wcet(V2),wcet(V3))

Loop: wcet(for V1 loop V2) =
(n+1)*wcet(V1) + n*wcet(V2)

Basic block: wcet(V) = $Time_V$

Optimal selection: ILP formulation

- Objective function: *minimize WCET of root*
- Convert timing rules for interior nodes to linear constraints

Sequence: wcet(V1,…,Vn) =
wcet(V1) +… + wcet(Vn)

$$wcet_V = \sum_{i=1}^{k} wcet_{V_i}$$

Branch: wcet(if V1 then V2 else V3) =
wcet(V1)+max(wcet(V2),wcet(V3))

$$wcet_V \geq wcet_{V_1} + wcet_{V_3}$$
$$wcet_V \geq wcet_{V_1} + wcet_{V_2}$$

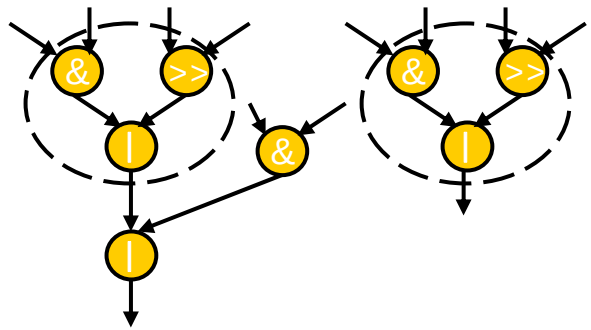
Loop: wcet(for V1 loop V2) =
(n+1)*wcet(V1) + n*wcet(V2)

$$wcet_V = (n+1) \times wcet_{V_1} + n \times wcet_{V_2}$$

Basic block: wcet(V) = $Time_V$

Execution time of basic block

- Boolean variables:
  - S_i = 1 if pattern i is selected
  - k_i = 1 if pattern instance i is selected

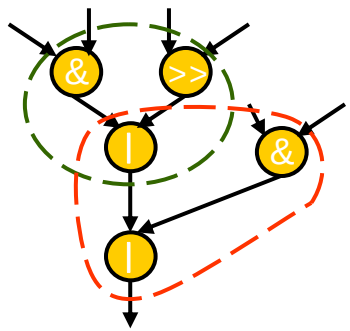


- Execution time of basic block

$$wcet_V = Time_V - \sum_{x \in V} k_x \times P_x$$

- At most one pattern instance covers an operation

$$\sum_{y \in Y} k_y \leq 1$$



Global constraints

- Resource constraint

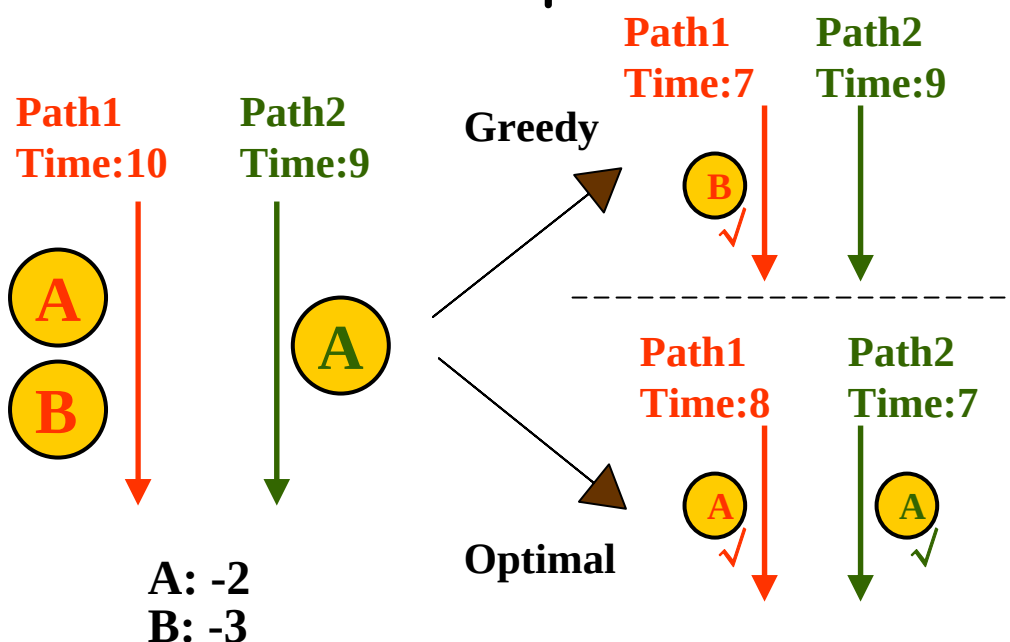
$$\sum_{i=1}^{N} (S_i \times R_i) \leq R$$

- Number of custom instructions constraint

$$\sum_{i=1}^{N} S_i \leq M$$

Heuristic selection method

- ILP solution is not scalable
  - Commercial ILP solver takes more than a day with large number of pattern instances
- Heuristic solution: Optimizing current WCET path may result in local optima

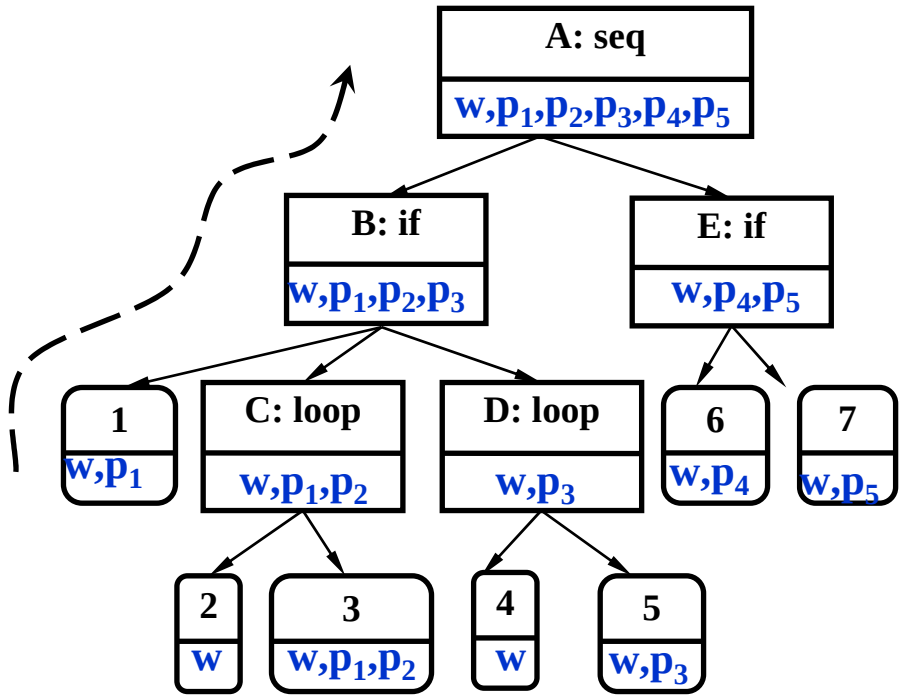


Basic heuristic

- Compute the global reduction in WCET, called *profit*, for each pattern
  - Not based on only the current WCET path
- Choose the pattern *p* with maximum profit
- Select the instances of pattern *p* that do not overlap with previously selected patterns
- Update the profit values for the remaining patterns

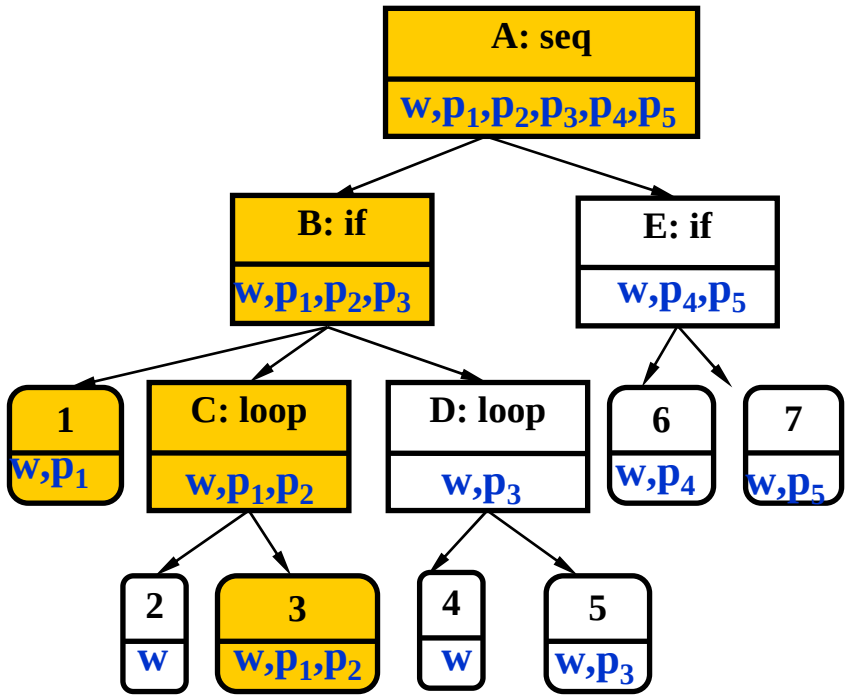
Computing profits

- Through a single bottom-up traversal of the syntax tree
- Annotate each interior node with
  - Profit values for all patterns appearing under it
  - WCET of the node
- Compute profit at an interior node by applying rules similar to timing schema



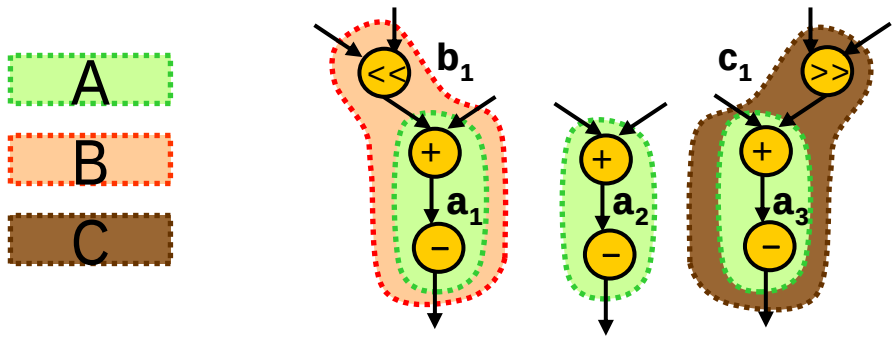
Updating profit

- Why update?
  - Selection of a pattern instance may preempt the selection of certain pattern instances
- Efficient update
  - Only update along the leaf to root node of basic blocks with selected pattern instances in the current round



Limitations of basic heuristic

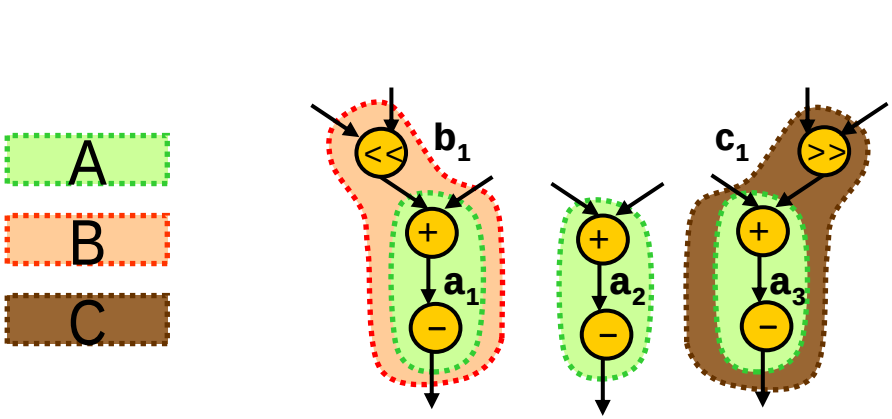
- Favors subsumed patterns

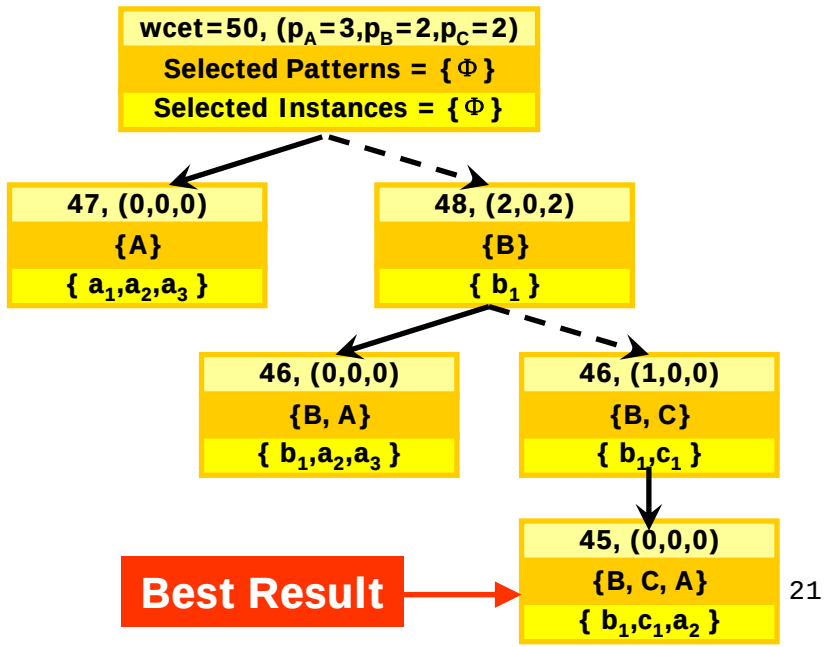


- $profit_A$: 3 $profit_B$: 2 $profit_C$: 2
- Basic Heuristic chooses A and all its instances
  - WCET improvement = 3
  - Preempts B and C
- Optimal: a2, b1, and c1
  - WCET improvement = 5

Improved Heuristic

- If best pattern *A* is subsumed by other patterns
  - either select *A*
  - or select subsuming pattern with max profit
- Choose the best result from these searches





Experimental Results

2 reg 1 imm inputs; 1 output; 5 custom instructions

Program	Ptn.	Inst.	WCET Red.		Time (s)	
			Heu	Opt.	Heu	Opt.
Adpcm	51	150	9%	9%	0.002	0.02
Blowfish	15	276	16%	16%	0.002	0.02
Compress	37	92	2%	2%	0.002	0.01
Crc	12	23	15%	15%	0.001	0.01
Djpeg	64	485	7%	7%	0.017	0.12
Gsmdec	158	2312	21%	22%	0.031	0.10
G721dec	73	180	4%	4%	0.006	0.03
Ndes	22	77	10%	10%	0.002	0.12
Rijndael	49	2520	16%	16%	0.034	1.25
Sha	9	40	12%	12%	0.001	0.01

4 reg/imm inputs; 2 outputs; 5 custom instructions

Program	Ptn.	Inst.	WCET Red.		Time (s)	
			Heu	Opt.	Heu	Opt.
Adpcm	101	258	14%	14%	0.005	0.04
Blowfish	56	1221	39%	39%	0.012	11.1
Compress	141	248	6%	6%	0.003	0.01
Crc	24	39	17%	17%	0.001	0.30
Djpeg	226	1056	11%	11%	0.028	0.28
Gsmdec	796	6782	26%	26%	0.064	0.05
G721dec	220	392	11%	11%	0.010	0.03
Ndes	77	182	17%	18%	0.003	0.03
Rijndael	156	9032	39%	39%	0.096	913
Sha	47	148	31%	31%	0.002	0.01

Significant WCET improvement under both typology
Improved heuristic achieves optimal most of the time

Experimental Results

2 reg 1 imm inputs; 1 output; 5 custom instructions

Program	Ptn.	Inst.	WCET Red.		Time (s)	
			Heu	Opt.	Heu	Opt.
Adpcm	51	150	9%	9%	0.002	0.02
Blowfish	15	276	16%	16%	0.002	0.02
Compress	37	92	2%	2%	0.002	0.01
Crc	12	23	15%	15%	0.001	0.01
Djpeg	64	485	7%	7%	0.017	0.12
Gsmdec	158	2312	21%	22%	0.031	0.10
G721dec	73	180	4%	4%	0.006	0.03
Ndes	22	77	10%	10%	0.002	0.12
Rijndael	49	2520	16%	16%	0.034	1.25
Sha	9	40	12%	12%	0.001	0.01

4 reg/imm inputs; 2 outputs; 5 custom instructions

Program	Ptn.	Inst.	WCET Red.		Time (s)	
			Heu	Opt.	Heu	Opt.
Adpcm	101	258	14%	14%	0.005	0.04
Blowfish	56	1221	39%	39%	0.012	11.1
Compress	141	248	6%	6%	0.003	0.01
Crc	24	39	17%	17%	0.001	0.30
Djpeg	226	1056	11%	11%	0.028	0.28
Gsmdec	796	6782	26%	26%	0.064	0.05
G721dec	220	392	11%	11%	0.010	0.03
Ndes	77	182	17%	18%	0.003	0.03
Rijndael	156	9032	39%	39%	0.096	913
Sha	47	148	31%	31%	0.002	0.01

Improved heuristic is much faster compared to commercial LP solver

Experimental Results

2 reg 1 imm inputs; 1 output; 5 custom instructions

Program	Ptn.	Inst.	WCET Red.		Time (s)	
			Heu	Opt.	Heu	Opt.
Adpcm	51	150	9%	9%	0.002	0.02
Blowfish	15	276	16%	16%	0.002	0.02
Compress	37	92	2%	2%	0.002	0.01
Crc	12	23	15%	15%	0.001	0.01
Djpeg	64	485	7%	7%	0.017	0.12
Gsmdec	158	2312	21%	22%	0.031	0.10
G721dec	73	180	4%	4%	0.006	0.03
Ndes	22	77	10%	10%	0.002	0.12
Rijndael	49	2520	16%	16%	0.034	1.25
Sha	9	40	12%	12%	0.001	0.01

4 reg/imm inputs; 2 outputs; 5 custom instructions

Program	Ptn.	Inst.	WCET Red.		Time (s)	
			Heu	Opt.	Heu	Opt.
Adpcm	101	258	14%	14%	0.005	0.04
Blowfish	56	1221	39%	39%	0.012	11.1
Compress	141	248	6%	6%	0.003	0.01
Crc	24	39	17%	17%	0.001	0.30
Djpeg	226	1056	11%	11%	0.028	0.28
Gsmdec	796	6782	26%	26%	0.064	0.05
G721dec	220	392	11%	11%	0.010	0.03
Ndes	77	182	17%	18%	0.003	0.03
Rijndael	156	9032	39%	39%	0.096	913
Sha	47	148	31%	31%	0.002	0.01

Improved heuristic is much faster compared to commercial LP solver

Conclusion

- Custom instructions can help us to meet deadlines
- WCET-guided efficient methodology for selection of custom instructions

Thank You