A generic framework for blackbox components in WCET computation

C. BALLABRIGA, H. CASSE, M. DE MICHIEL

{ballabri, casse, michiel}@irit.fr

TRACES – IRIT - Université de Toulouse - FRANCE

PLAN

introduction

examples of partial analyses

generalization of the approach

experimentation with OTAWA

conclusion

WCET computation*

WCET computation by static analysis

program control flow analysis

architecture effects analysis

WCET computation → IPET

IPET (Implicit Path Enumeration Technique)

WCET = maximization of $t_i \times x_i$

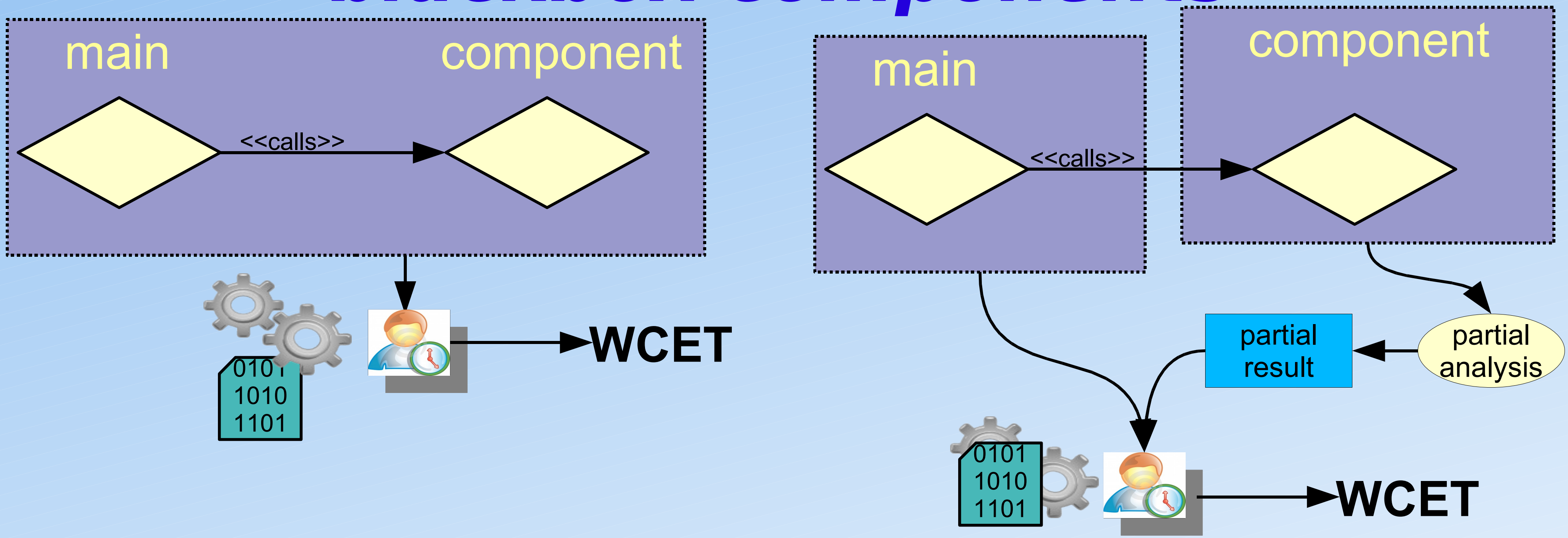
under constraints of an ILP system

- program control flow
- hardware effects

ILP solver

* Worst Case Execution Time

blackbox components



WCET computation done on a whole program
program = multiple components
lack of informations on blackbox components
→ no computation
how to handle blackbox component ?
partial analysus

* Worst Case Execution Time

PLAN

introduction

examples of partial analyses

generalization of the approach

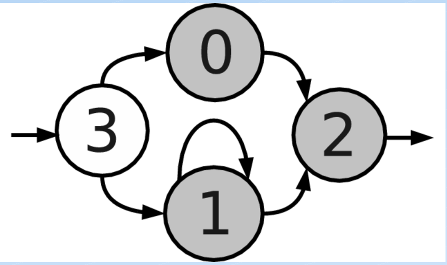
experimentation with OTAWA

conclusion

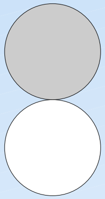
the instruction cache analysis

```
#define ERR -1
#define OK 0
int status = OK
int fact (int y) {
  int result = 1;
  int i;
  if (y >= 0) {
    for (i = 1; i <= y; i++)
      result *= i;
    status = OK;
  } else status = ERR;
  return result;
}
```

example program



CFG



BB in the current cache row

BB in other row

transfer function

$\text{state}_{\text{after}} = \textit{transfer}(\text{state}_{\text{before}})$

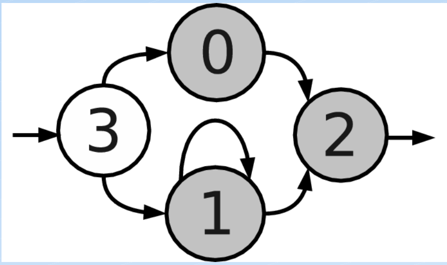
summary function

$\text{categories} = \textit{summary}(\text{state}_{\text{before}})$

the loop bounds estimation

```
#define ERR -1
#define OK 0
int status = OK
int fact (int y) {
  int result = 1;
  int i;
  if (y >= 0) {
    for (i = 1; i <= y; i++)
      result *= i;
    status = OK;
  } else status = ERR;
  return result;
}
```

example program



CFG

transfer function

$\text{state}_{\text{after}} = \textit{transfer}(\text{state}_{\text{before}})$

summary function

$\text{bounds} = \textit{summary}(\text{state}_{\text{before}})$

PLAN

introduction
examples of partial analyses
generalization of the approach
experimentation with OTAWA
conclusion

content of the partial result

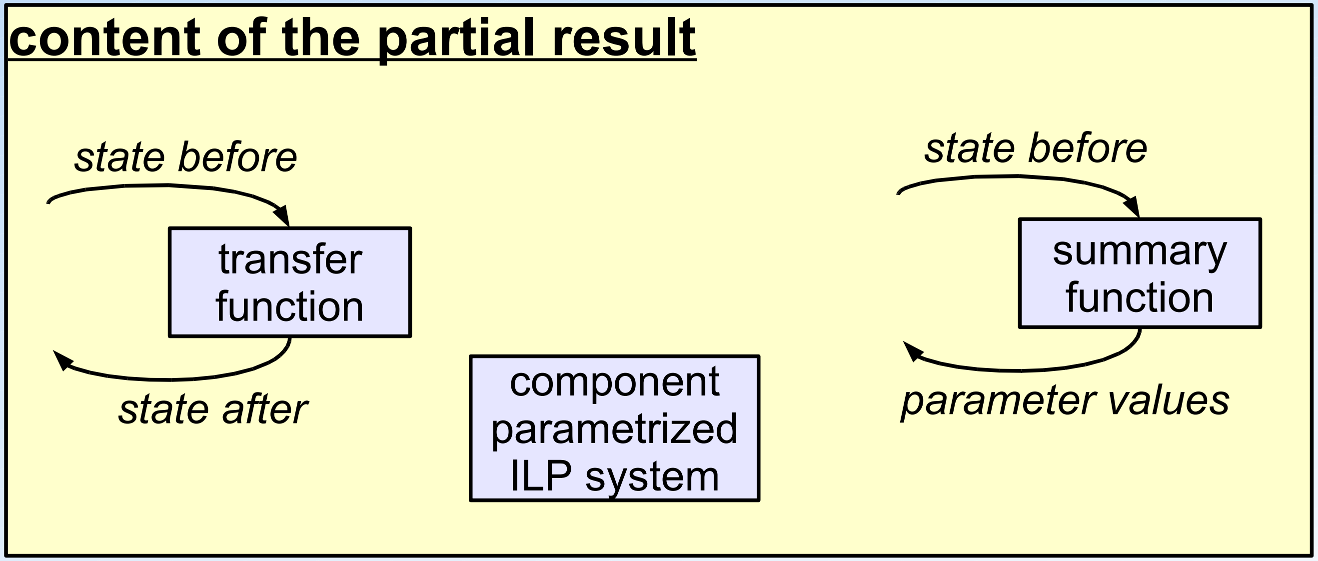
for each analysis

effect : component → main program = transfer function for each

effect : calling context → ILP parameters = summary function

for the component

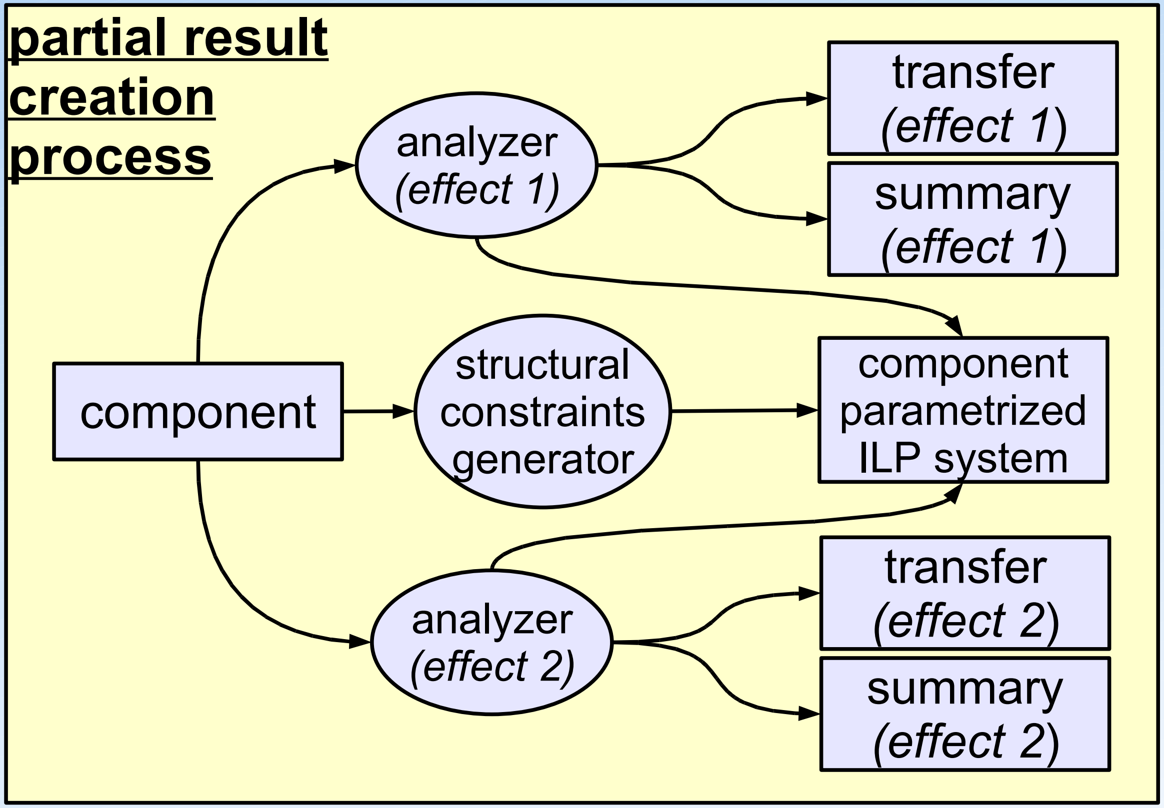
contribution of component = parametrized ILP system



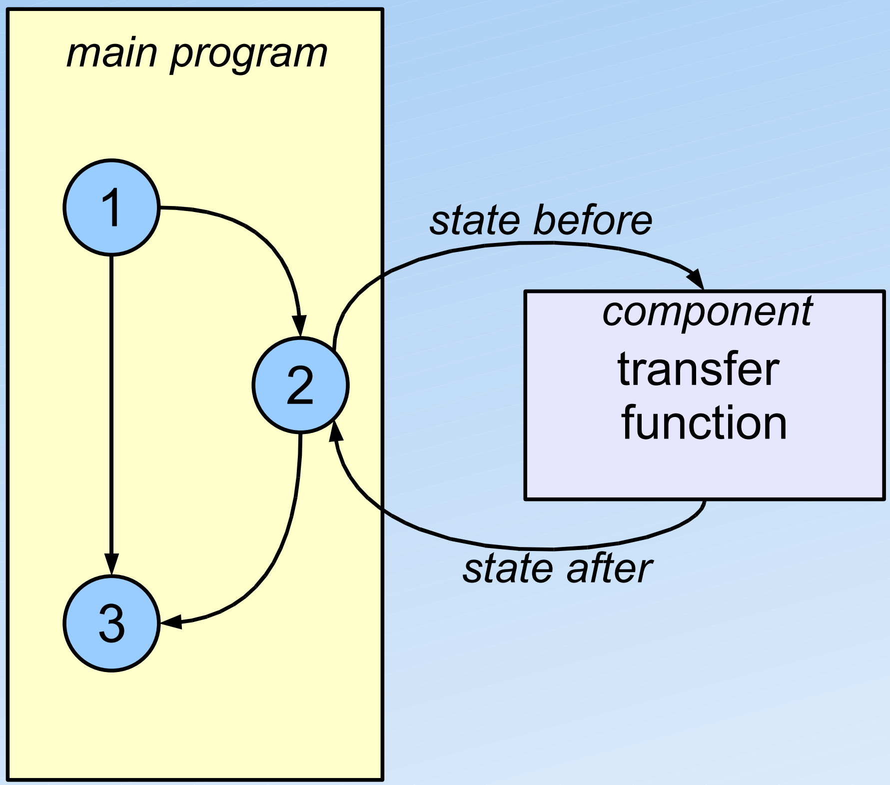
(1) creation of the partial result

each analyzer → transfer
→ summary

all analyzers → one ILP system



(2) usage of the partial result



<u>with the *transfer* function</u>

do each analysis on the main program

get the component entry state for each analysis

(2) usage of the partial result

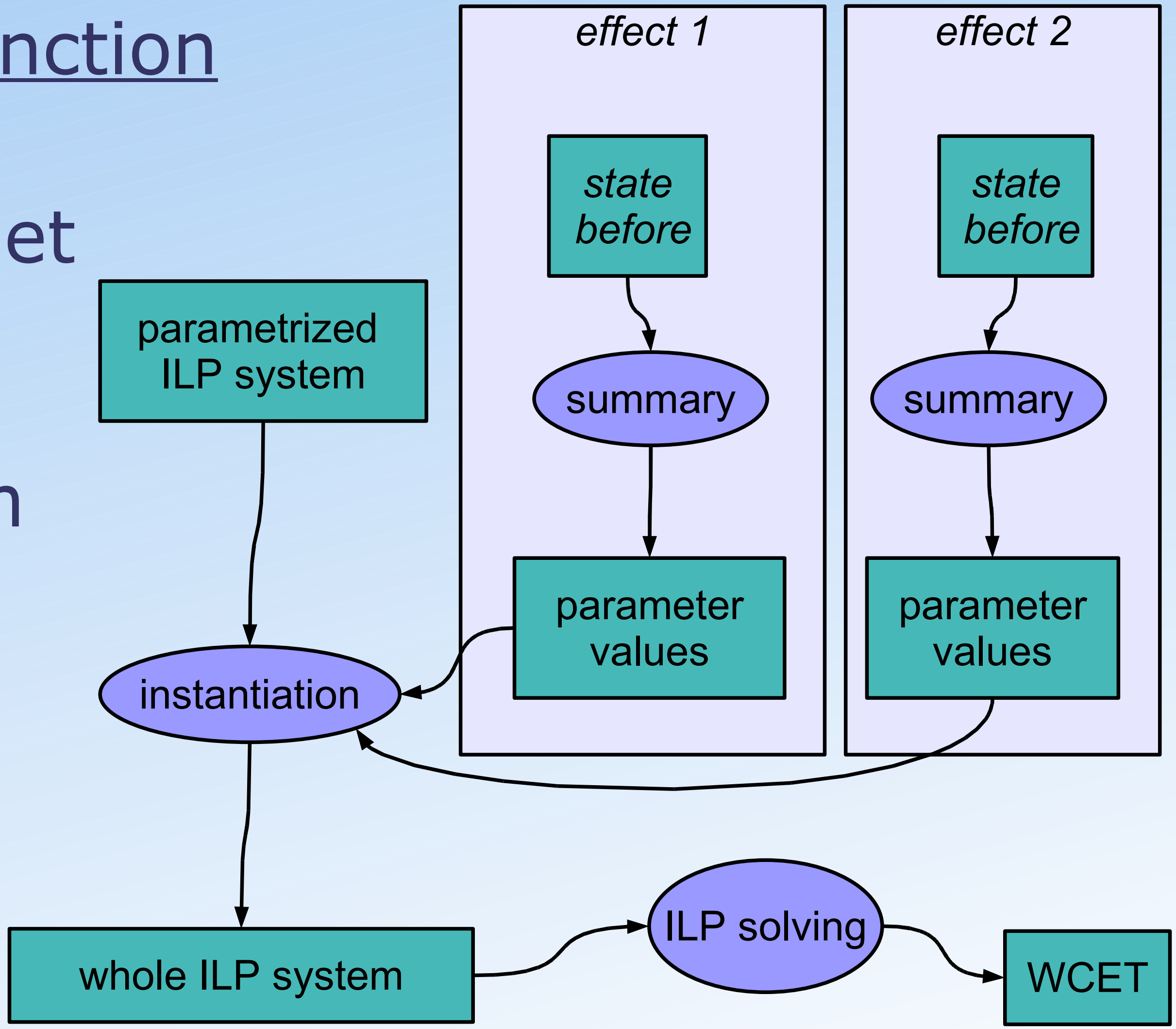
with the summary function

apply summaries to get parameter values

instantiate ILP system

solve ILP system

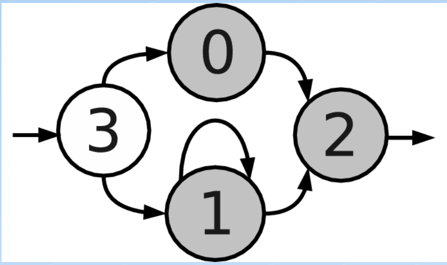
summary



partial result example

```
#define ERR -1
#define OK 0
int status = OK
int fact (int y) {
  int result = 1;
  int i;
  if (y >= 0) {
    for (i = 1; i <= y; i++)
      result *= i;
    status = OK;
  } else status = ERR;
  return result;
}
```

example program



CFG

parametrized ILP system

if (p_0=Always_Hit) $x^{miss}_0 = 0$
if (p_1=Always_Hit) $x^{miss}_1 = 0$
if (p_2=Always_Hit) $x^{miss}_2 = 0$
$e_{1,1} = p_3 \cdot e_{3,1}$

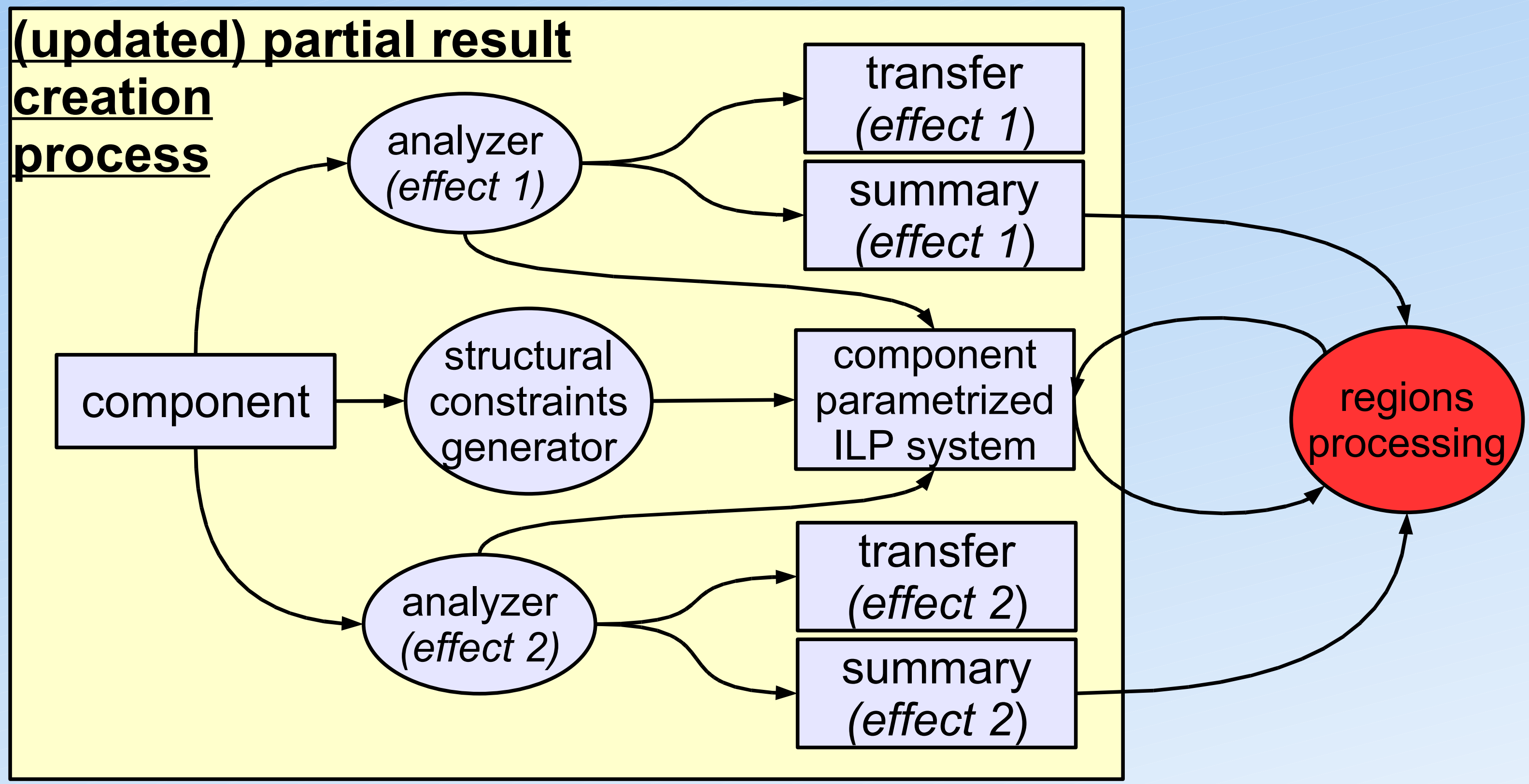
transfer

cache transfer:
ageing (must)
block 0 ⇒ aged 2 times
block 1 ⇒ aged 2 times
inserted blocks (must)
block 2 ⇒ inserted at age 0
loop bounds transfer:
$\{x = x + 10\}$

summary

cache summary:
CacheBlock 0, parameter p_0=
if must(0) < 4 ⇒ Always Hit
else Not Classified
CacheBlock 1, parameter p_1=
if must(1) < 4 ⇒ Always Hit
else Not Classified
CacheBlock 2, parameter p_2=
if must(2) < 3 ⇒ Always Hit
else Not Classified
loop bounds summary:
Loop 1, parameter p_3= 3y

Precomputing independent regions



general idea:

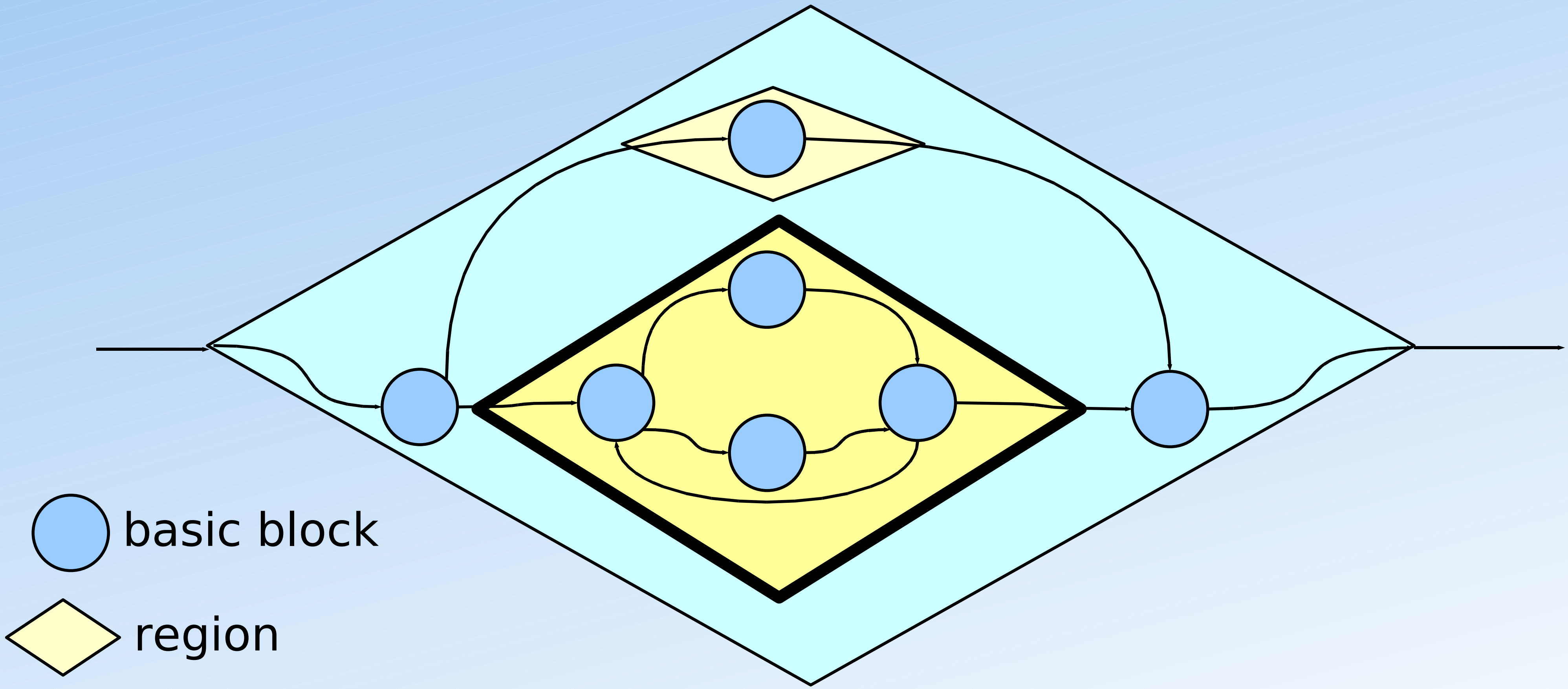
pre-compute the WCET of context-independent parts of the component to produce a smaller ILP system.

(we need the summary functions to know which parts are context-independent)

Precomputing independent regions

Single-Entry Single-Exit regions

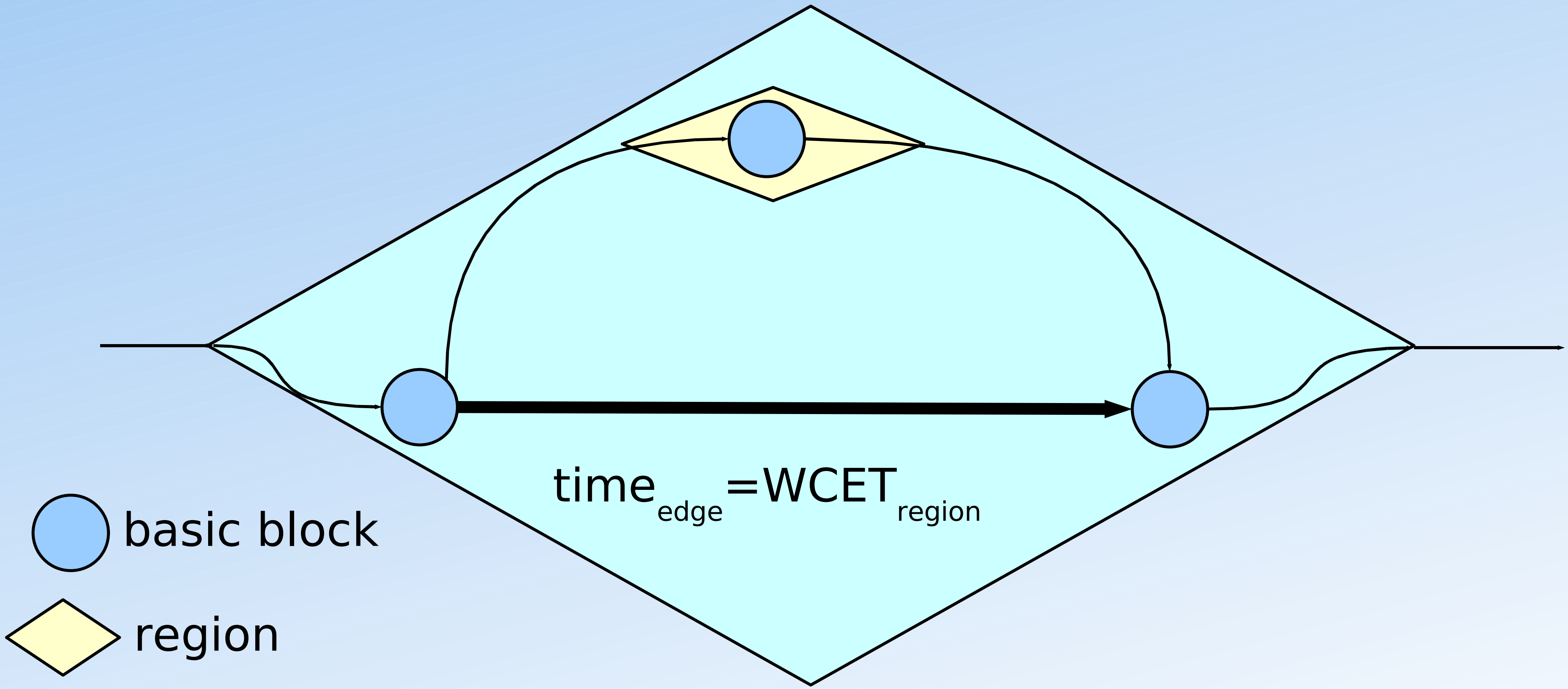
Constant-WCET regions can be pre-computed



Precomputing independent regions

Single-Entry Single-Exit regions

Constant-WCET regions can be pre-computed



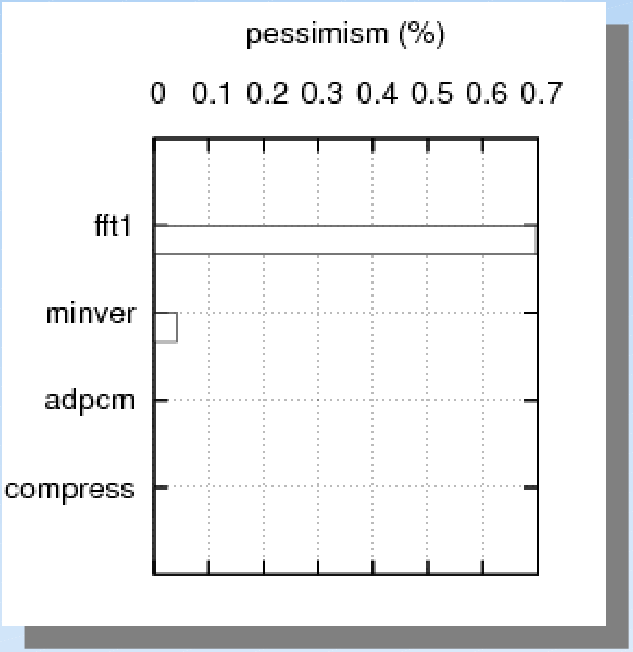
(the region is now modeled by a single edge)

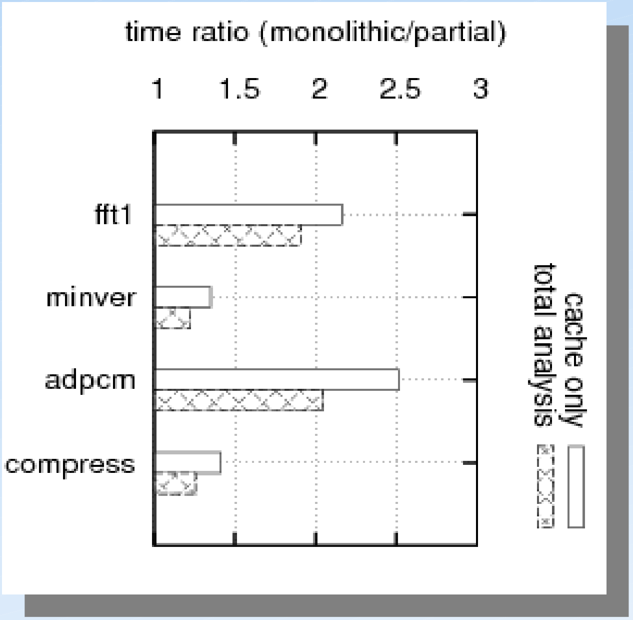
PLAN

introduction
examples of partial analyses
generalization of the approach
experimentation with OTAWA
conclusion

experimentation with OTAWA

pessimism increase measurement
average WCET increase: 0.18**%**





analysis time measurement
average gain: 1.86 **times**

PLAN

introduction
examples of partial analyses
generalization of the approach
experimentation with OTAWA
conclusion

Conclusion and future works

blackbox components analysis

- handle COTS correctly
- speed up WCET computation

future works

- finishing branch predictor partial analysis
- automate summary/transfer function creation