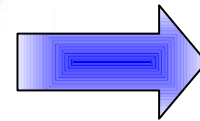
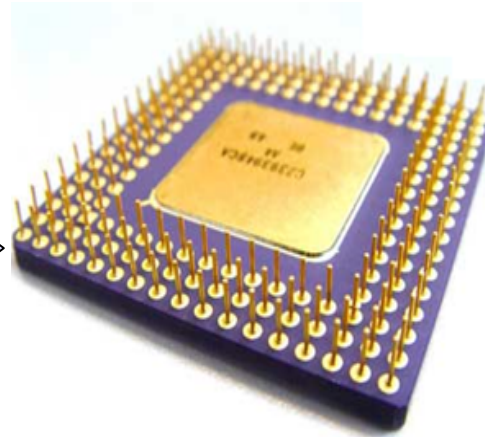


SMT-based Test Program Generation for Cache Memory

Evgeny Kornikhin
Moscow State University

Testing by Test Programs

```
add r1,r2,r3  
sub r4, r1, r2  
lw r5, r1, 0  
lui r2, r1, r4
```



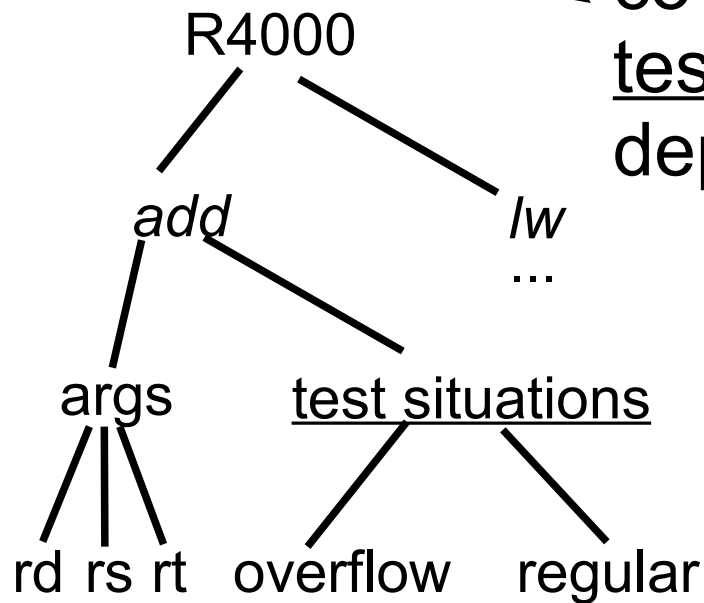
Y/N

assembler program
(test program)

microprocessor

Test Programs Generation

model of microprocessor

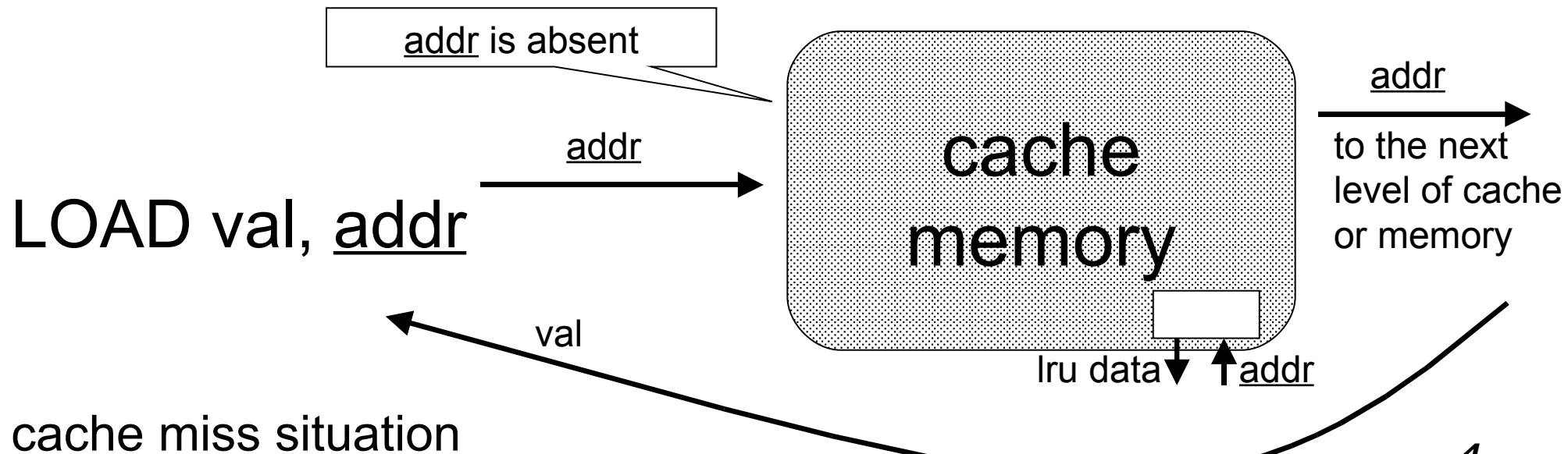
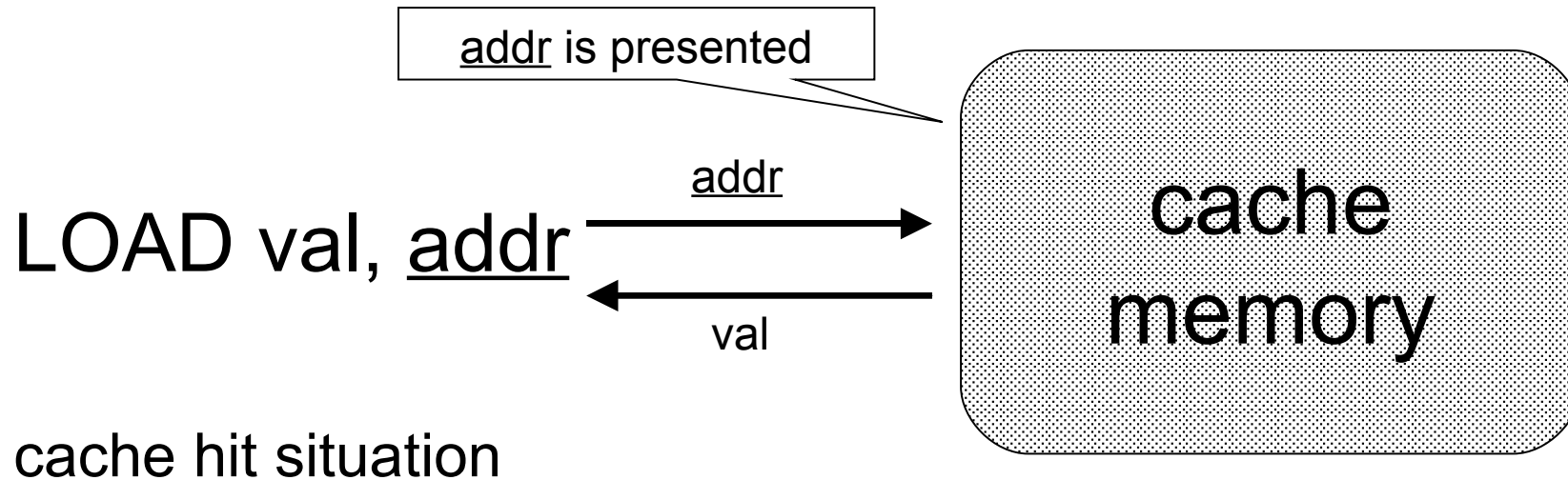


coverage of
test situations and
dependencies (RAW, RAR)

add r1,r2,r3 @ overflow
lw r4, r3, c @ hit
test templates
(abstract form)

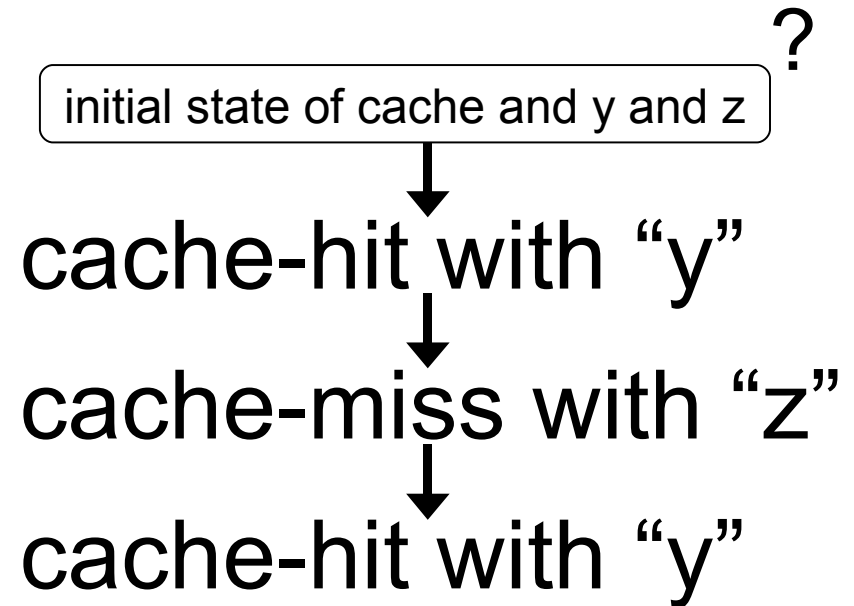
mov r2, 0xFF
add r1,r2,r3
lw r4, r3, 0
test programs
(executable form)

Cache memory behavior



Problem again

LOAD x, y @ hit
STORE u, z @ miss
LOAD z, y @ hit



initial state of cache
(addresses of cached data)
and registers = ?

Huge exhaustive search

LOAD x, y @ hit

STORE u, z @ miss

LOAD z, y @ hit

select values
↓↑
run template

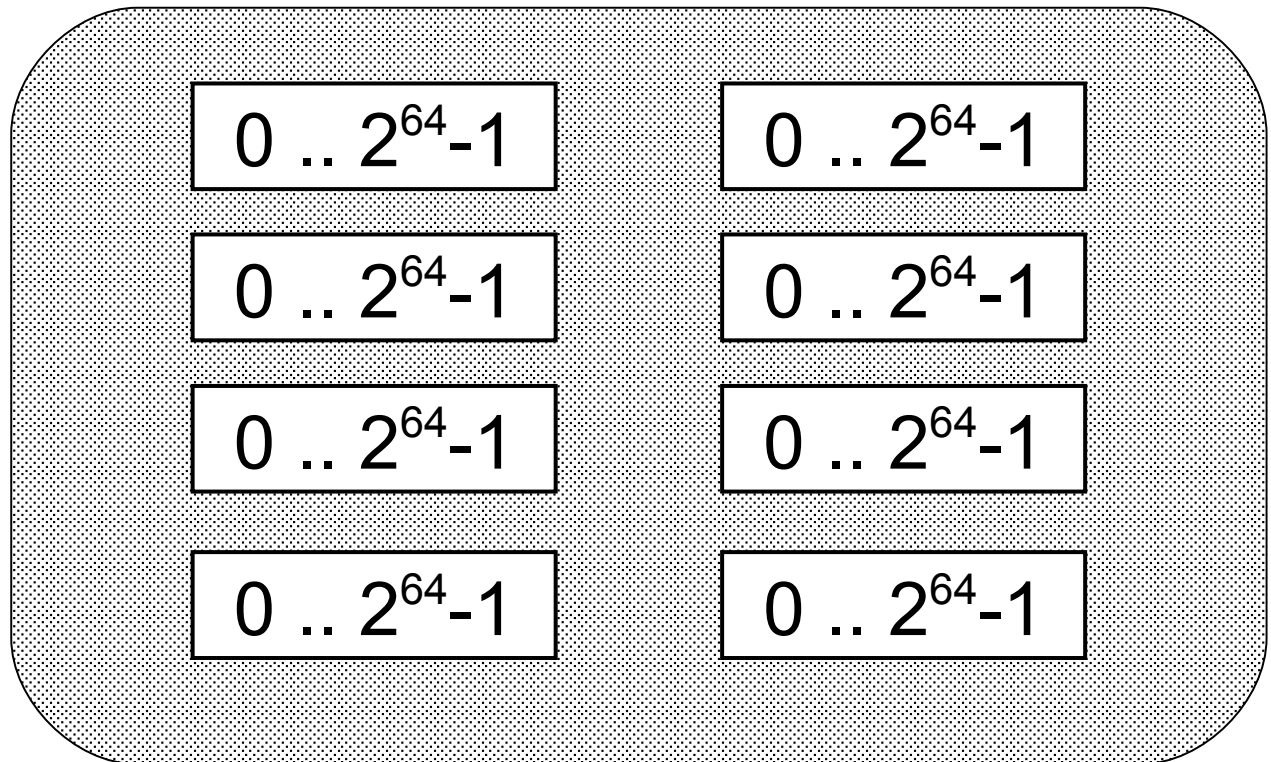
x $0 \dots 2^{64}-1$

y $0 \dots 2^{64}-1$

z $0 \dots 2^{64}-1$

u $0 \dots 2^{64}-1$

initial registers'
values



initial cache state

Hard behavior constraints

LOAD x, y @ hit
(cache-hit with “y”) $\left\{ \begin{array}{l} \text{index}(L, y) > 0 \\ L' == \text{inc_counter}(L, y) \end{array} \right.$

STORE u, z @ miss \rightarrow z0
(cache-miss with “z”) $\left\{ \begin{array}{l} \text{index}(L', z) == 0 \\ \text{counter}(L', z0) \rightarrow \text{min} \\ L'' == \text{remove}(L', z0) \\ L''' == \text{add}(L'', z) \end{array} \right.$

LOAD z, y @ hit
(cache-hit with “y”) $\left\{ \begin{array}{l} \text{index}(L''', y) > 0 \end{array} \right.$

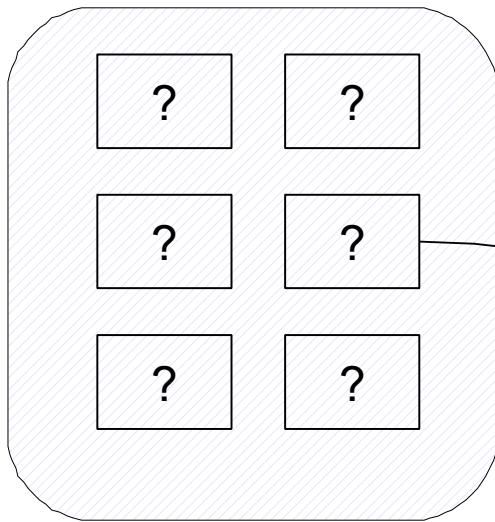
x, y, z, u, L = ?

Key Idea

test template

add ...
load ...
sub ...
div ...

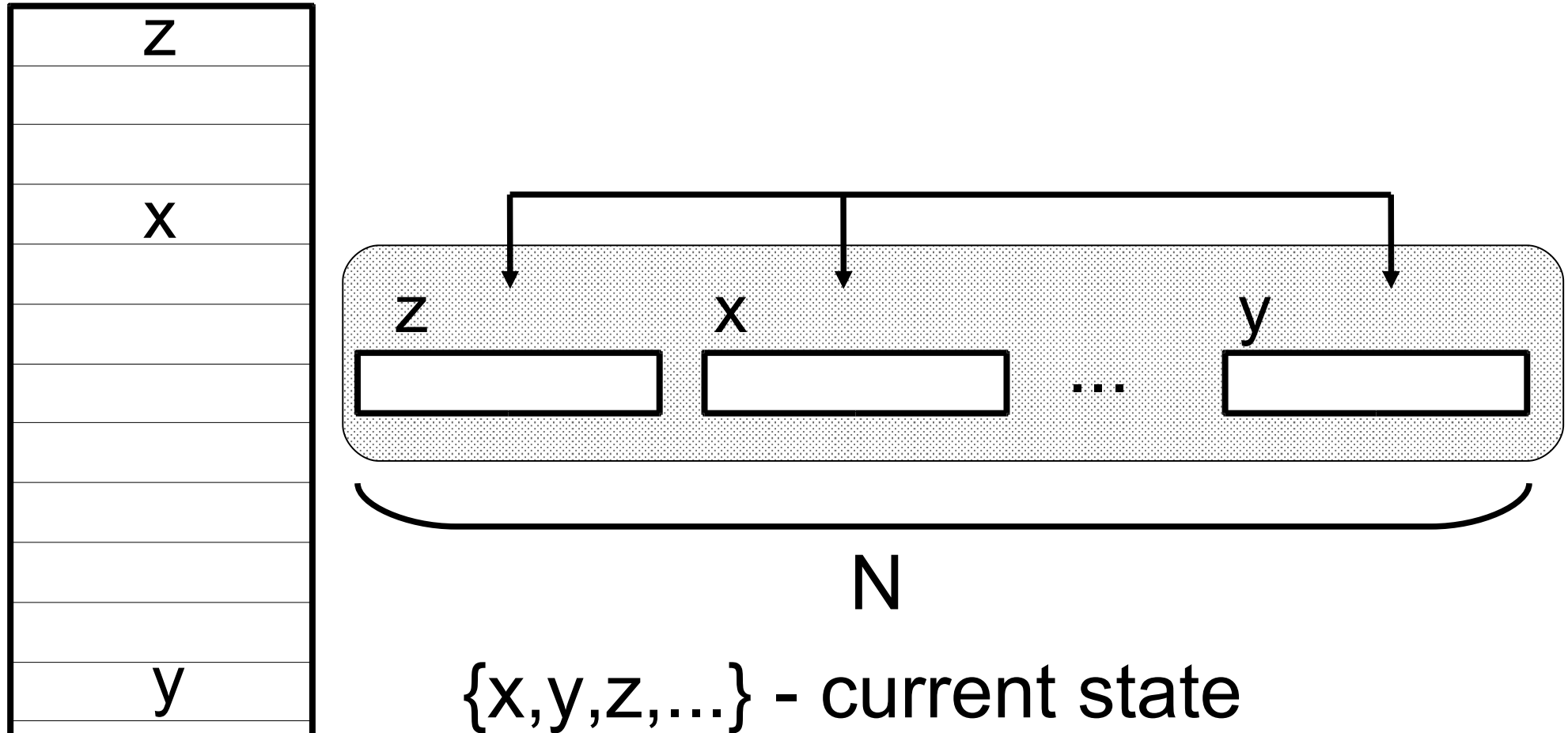
LOAD x, y @ hit



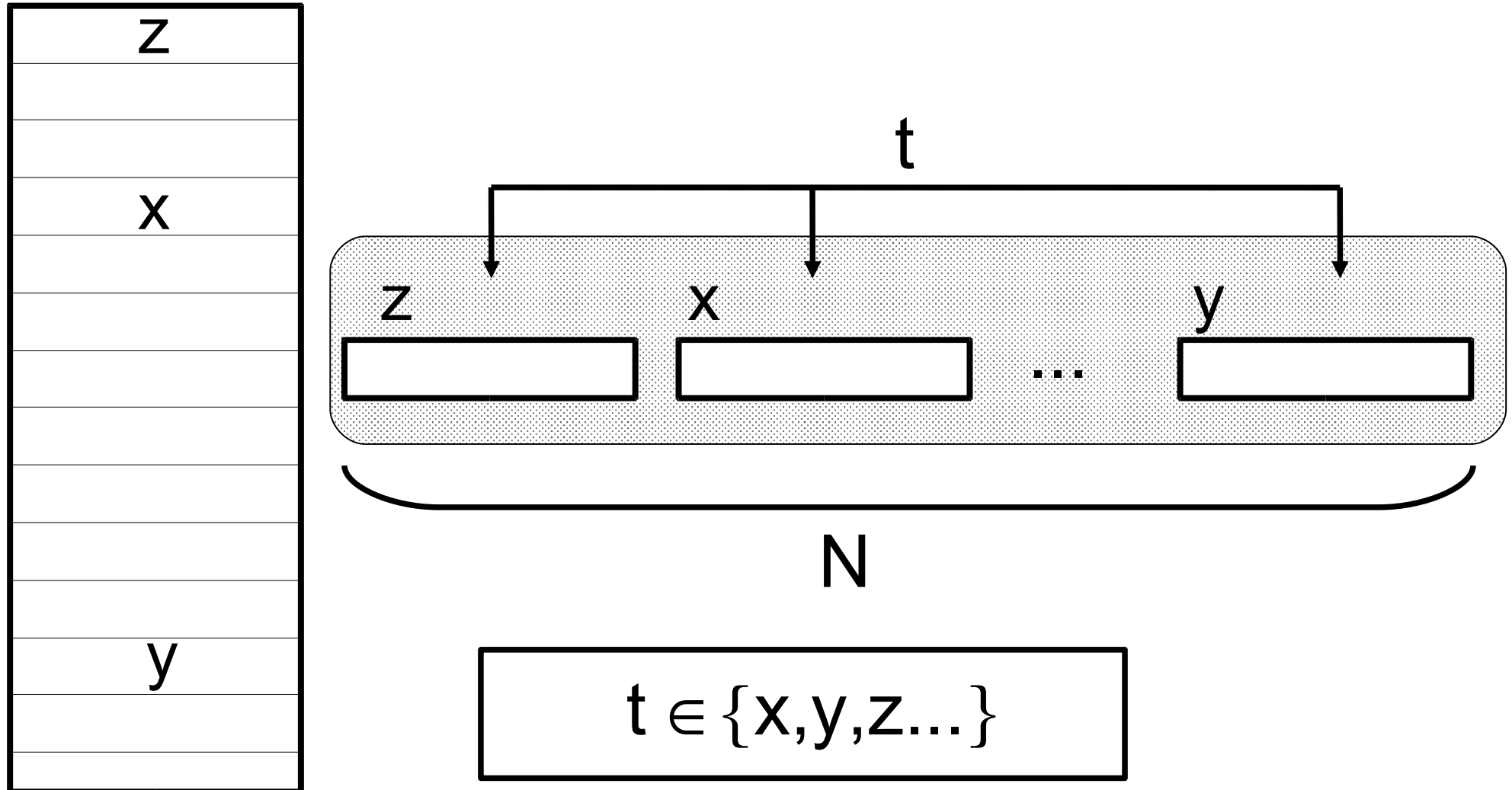
cache model

$y \in \{a, b, c\}$
 $u \notin \{a, b, c\}$
 $x = z$

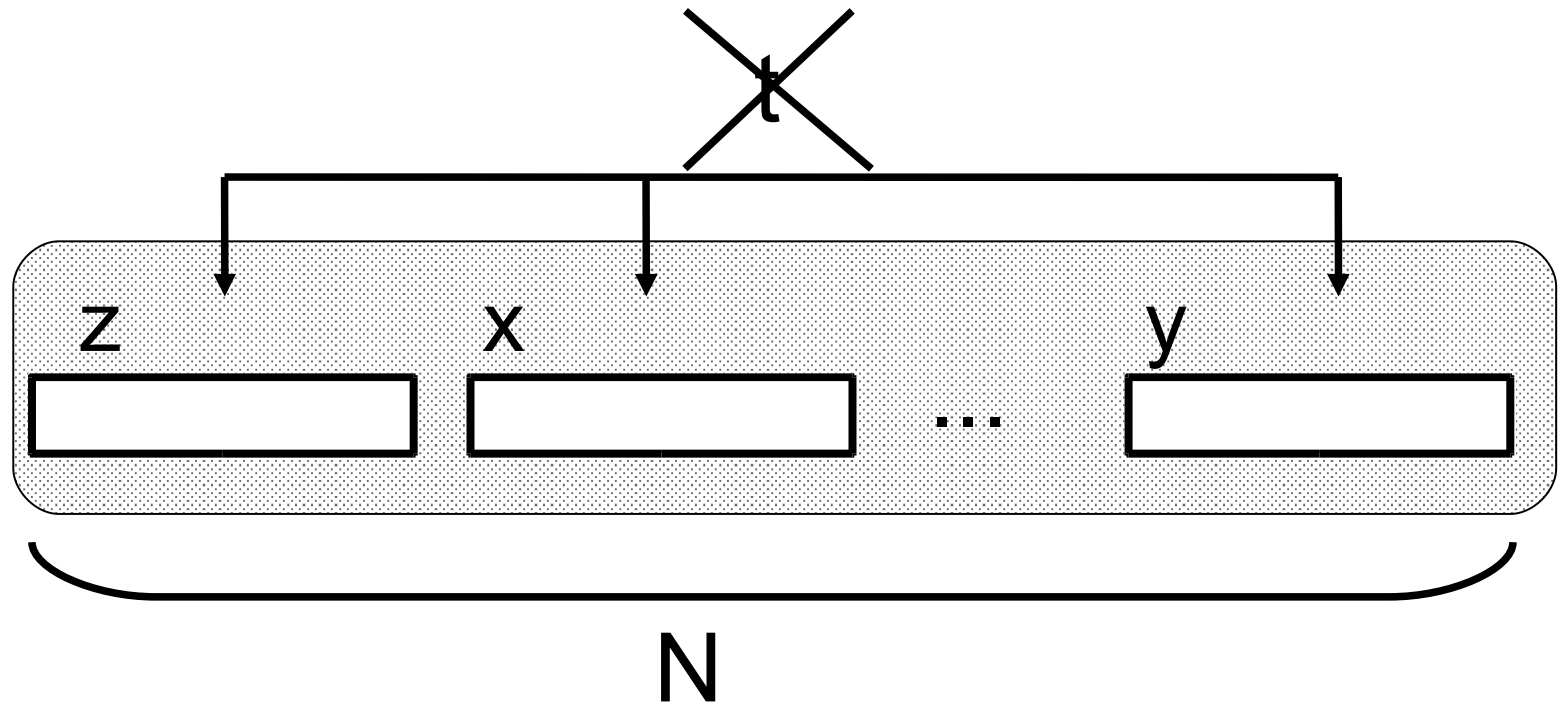
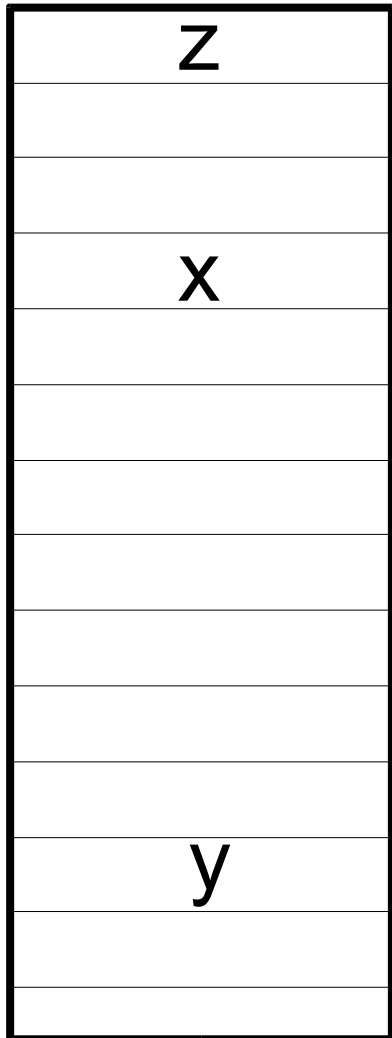
Fully Associative Cache



Cache-hit hit(t)



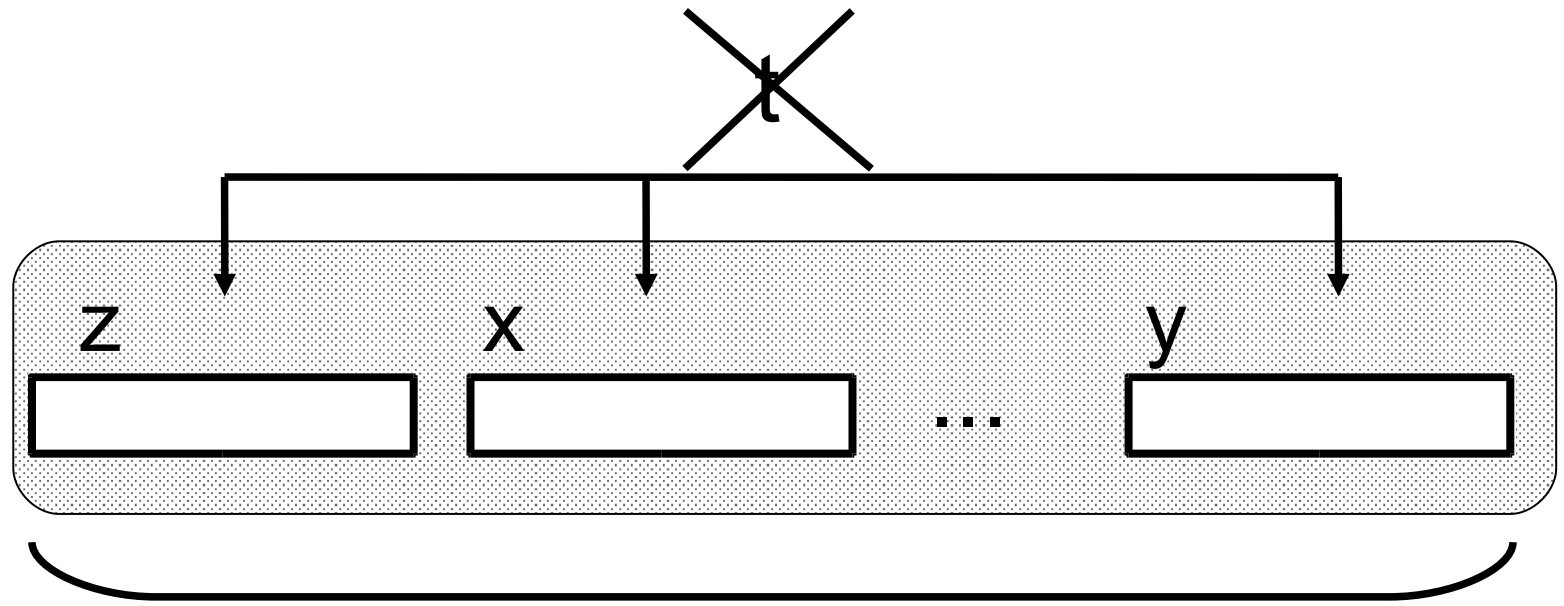
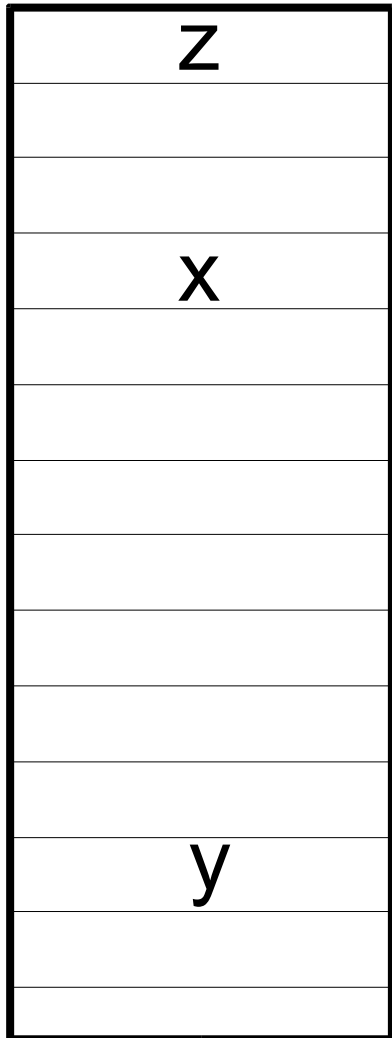
Cache-miss miss(t)



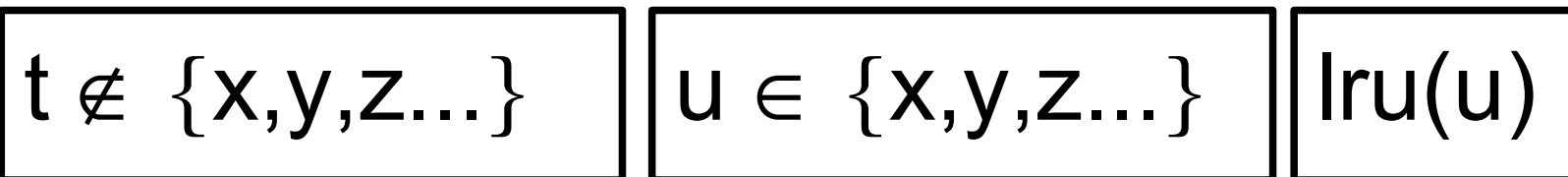
$t \notin \{x, y, z, \dots\}$

new cache $\{x, y, z, \dots\} \cup \{t\} \setminus \{?\}$

Cache-miss $\text{miss}(t) \rightarrow u$



N



new cache $\{x, y, z, \dots\} \cup \{t\} \setminus \{u\}$

$\text{lru}(u)$

hit x1

hit x2

miss $x3 \rightarrow x4$

hit x5

(L)

miss $t \rightarrow u$



$$\begin{cases} u = x2 \\ \{x3, x5\} = L \setminus \{u\} \end{cases}$$

$\text{lru}(u)$

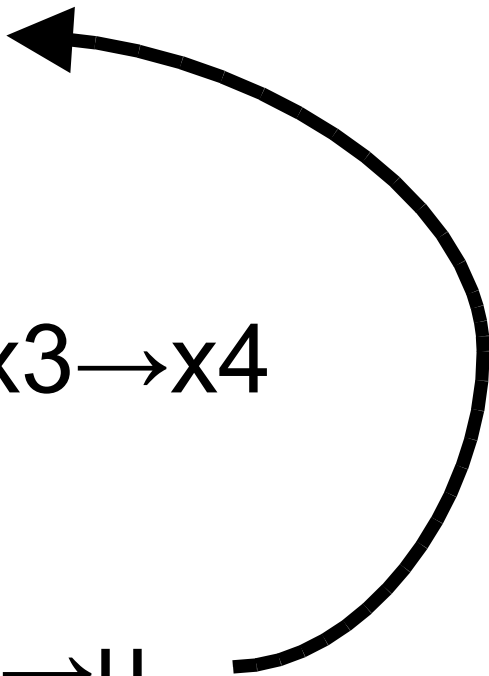
hit x1

hit x2

miss $x3 \rightarrow x4$

hit x5

Ⓛ miss $t \rightarrow u$



$$\begin{cases} u = x1 \\ \{x2, x3, x5\} = L \setminus \{u\} \end{cases}$$

there are another cases..

Example

initial
state:

α
 β
 γ

LOAD x, y @ hit

STORE u, z @ miss $\rightarrow z_0$

LOAD z, y @ hit

$N = 3$

$$y \in \{\alpha, \beta, \gamma\}$$

$$z \notin \{\alpha, \beta, \gamma\}$$

$$z_0 \in \{\alpha, \beta, \gamma\}$$

$$z_0 = \beta$$

$$\{\alpha, \beta, \gamma\} \setminus \{z_0\} = \{\gamma, y\}$$

$$y \in \{\alpha, \beta, \gamma, z\} \setminus \{z_0\}$$

Example

$$y = \alpha$$

$$z \notin \{\alpha, \beta, \gamma\}$$

$$y = \alpha = 0$$

$$\beta = 1$$

$$\gamma = 2$$

$$z = 3$$

$$y \in \{\alpha, \beta, \gamma\}$$

$$z \notin \{\alpha, \beta, \gamma\}$$

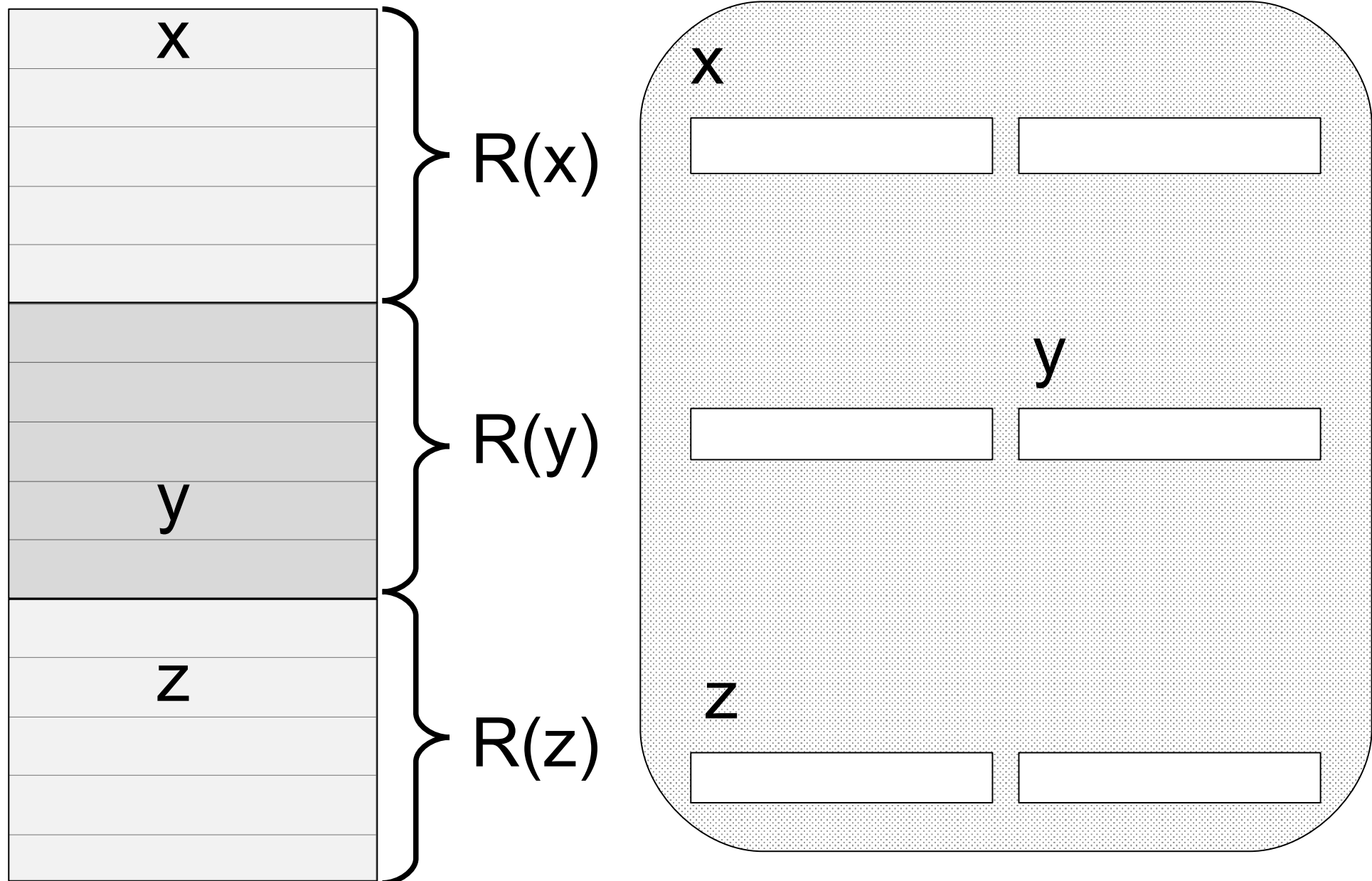
$$z_0 \in \{\alpha, \beta, \gamma\}$$

$$z_0 = \beta$$

$$\{\alpha, \beta, \gamma\} \setminus \{z_0\} = \{\gamma, y\}$$

$$y \in \{\alpha, \beta, \gamma, z\} \setminus \{z_0\}$$

Common Cache



Common Cache

hit(t)

$$t \in L$$

miss(t) \rightarrow u

$$u \in L$$

$$t \notin L$$

$L \cup \{t\} \setminus \{u\}$ is the next cache

$$R(t) = R(u)$$

$$lru(u)$$

$\text{lru}(u)$

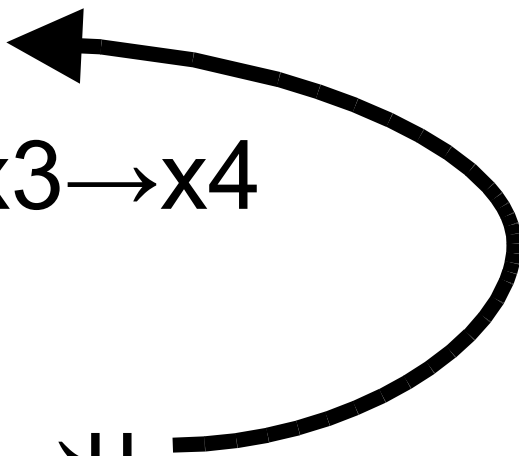
hit x1

hit x2

miss $x3 \rightarrow x4$

hit x5

miss $t \rightarrow u$



$$\begin{cases} u = x2 \\ \{x3, x5\} \cap R(u) \\ = (L \setminus \{y\}) \cap R(u) \end{cases}$$

Example

$$x_1, x_2 \in \{a_1, a_2, b_1, b_2, c_1, c_2\}$$

$$x_3 \notin \{a_1, a_2, b_1, b_2, c_1, c_2\}$$

$$R(x_3) = R(y_3)$$

$$x_4 \in \{a_1, \dots, c_2, x_3\} \setminus \{y_3\}$$

$$x_5 \notin \{a_1, \dots, c_2, x_3\} \setminus \{y_3\}$$

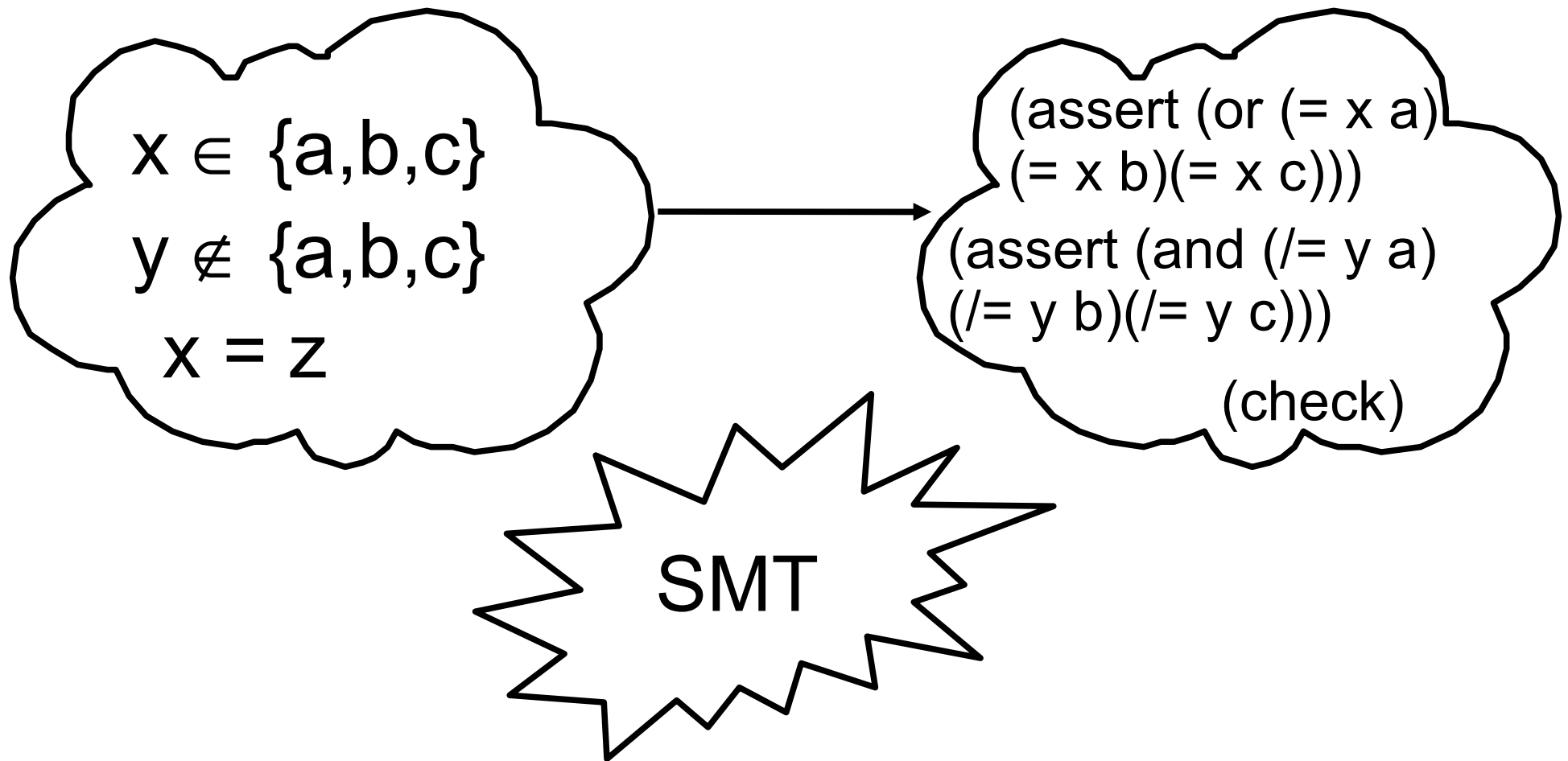
$$y_3 = c_2$$

$$\{y_3\} = (\{a_1, \dots, c_2\} \setminus \{x_1, x_2, y_3\}) \cap R(y_3)$$

$$y_5 = x_2$$

$$\{y_5\} = (\{a_1 \dots c_2, x_3\} \setminus \{y_3, y_5, x_3, x_4\}) \cap R(y_5)$$

Solver



SAT modulo theories

Yices

Conclusions

- Cache behavior can be expressed using equations on addresses' set
- Equations can be solved by SMT-solver



Contacts

<http://tesla-project.googlecode.com>

<http://hardware.ispras.ru>

kornevgen@gmail.com



Thank you for attention!