

Parallel Programming

Architectures

Prof. Paolo Bientinesi

HPAC, RWTH Aachen
`pauldj@aices.rwth-aachen.de`

WS17/18



On dependencies and parallel execution

- Two independent instructions can be executed
“concurrently” = “in parallel” (by different resources)

On dependencies and parallel execution

- Two independent instructions can be executed “concurrently” = “in parallel” (by different resources)
- Parallel execution: **NO** time ordering:
When two instructions are executed in parallel, **NO** assumptions about start, duration, end.

On dependencies and parallel execution

- Two independent instructions can be executed “concurrently” = “in parallel” (by different resources)
- Parallel execution: **NO** time ordering:
When two instructions are executed in parallel, **NO** assumptions about start, duration, end.
- Dependencies \Rightarrow ordering
The “right” ordering is dictated by the semantics of the program

On dependencies and parallel execution

- Two independent instructions can be executed “concurrently” = “in parallel” (by different resources)
- Parallel execution: **NO** time ordering:
When two instructions are executed in parallel, **NO** assumptions about start, duration, end.
- Dependencies \Rightarrow ordering
The “right” ordering is dictated by the semantics of the program
- Some dependencies can be removed by duplicating data

True / Flow dependency

$\{x = 1, y = 2, a = 3\}$

...

$y := a * x + y$

$w := 3 * y$

...

$\{y = 5, w = 15\}$

True / Flow dependency

```
{x = 1, y = 2, a = 3}
```

```
...
```

```
y := a * x + y
```

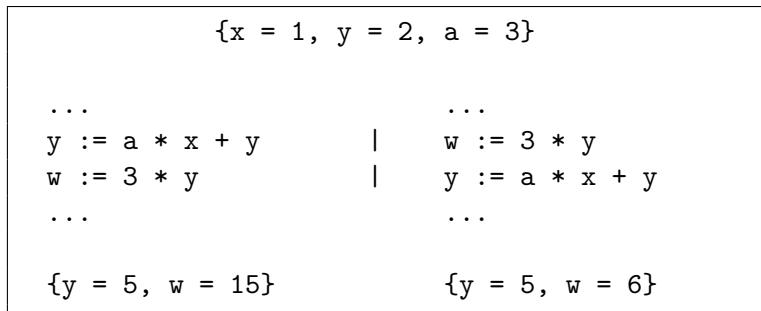
```
w := 3 * y
```

```
...
```

```
{y = 5, w = 15}
```

- The value of `w` depends on the updated value of `y`

True / Flow dependency



- The value of w depends on the updated value of y
- The semantics of the program depends on the **order** of the statements

Anti dependency

$\{x = 1, y = 2, a = 3\}$

...

$w := 3 * y$

$y := a * x + y$

...

$\{y = 5, w = 6\}$

- The value of w depends on the initial value of y

Anti dependency

$\{x = 1, y = 2, a = 3\}$

...		...
$w := 3 * y$		$y := a * x + y$
$y := a * x + y$		$w := 3 * y$
...		...
$\{y = 5, w = 6\}$		$\{y = 5, w = 15\}$

- The value of w depends on the initial value of y
- The semantics of the program depends on the **order** of the statements

Output dependency

$\{x = 1, y = 2, a = 3\}$

...

$w := 3 * y$

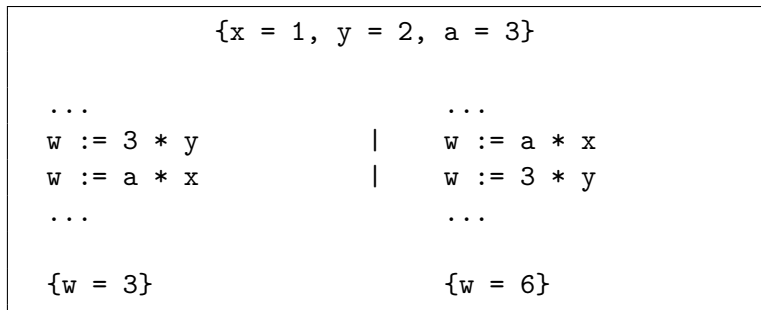
$w := a * x$

...

$\{w = 3\}$

- The value of w depends on the order of the statements

Output dependency



- The value of w depends on the order of the statements
- The semantics of the program depends on the **order** of the statements