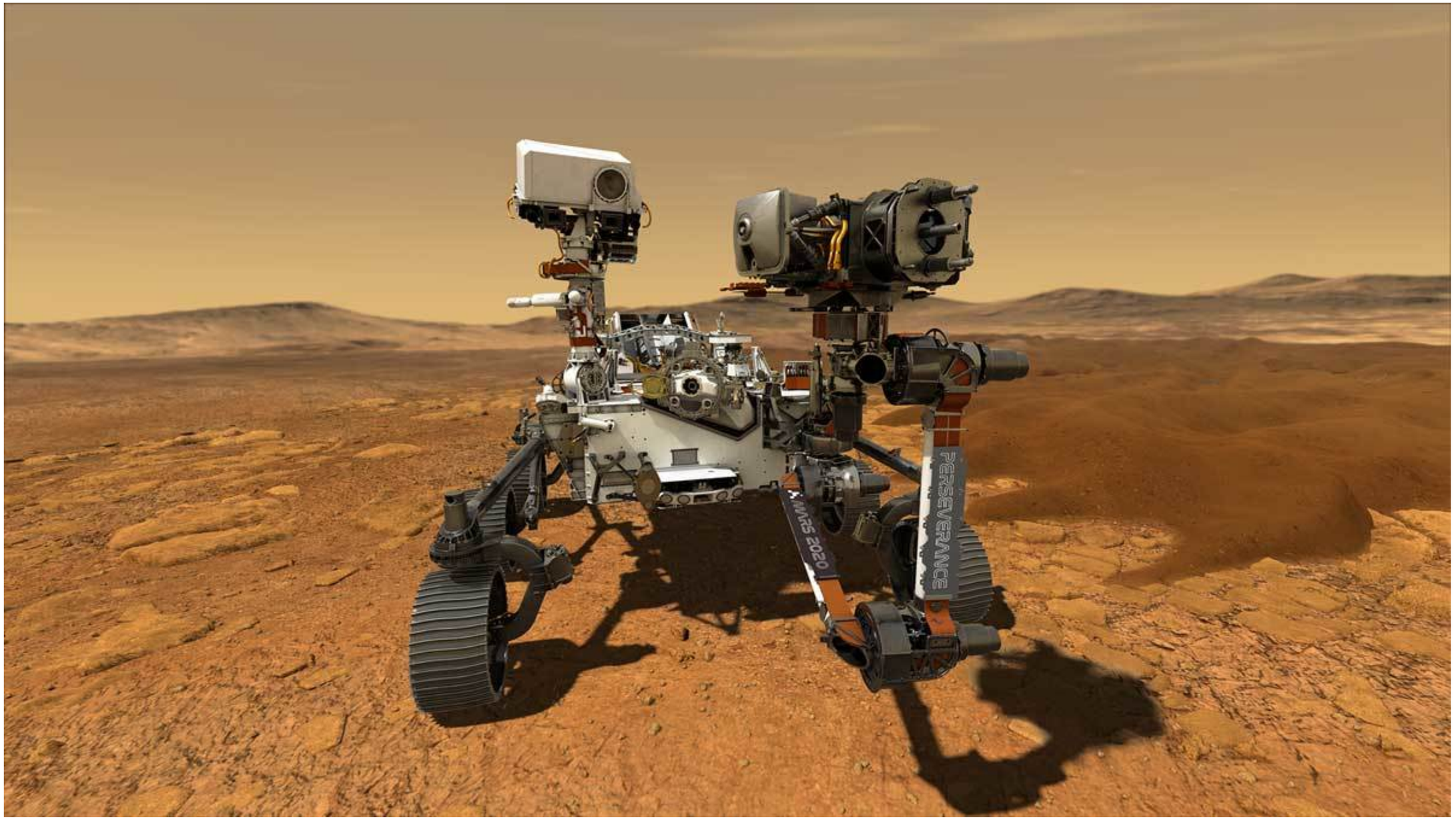
Impossible Made Possible

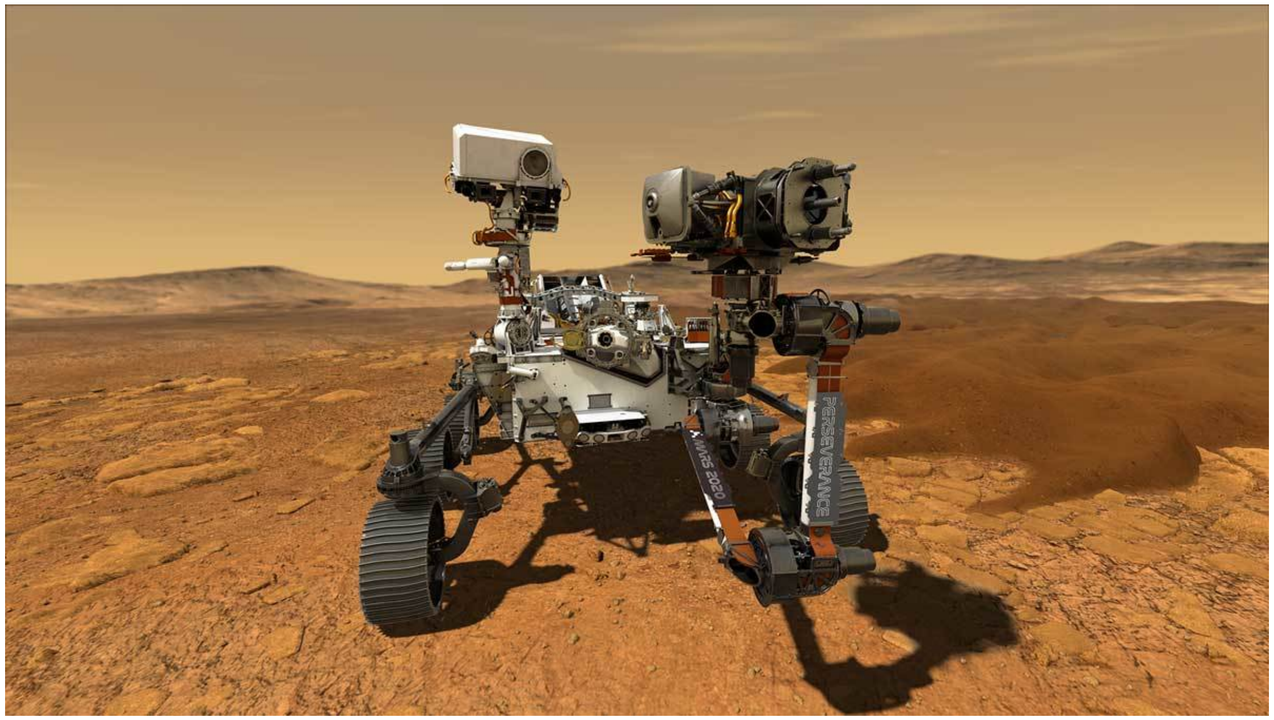
Encoding Intractable Specifications via Implied Domain Constraints

Chris Johannsen, Brian Kempa, Phillip H. Jones, Kristin Y. Rozier, Tichakorn Wongpiromsarn

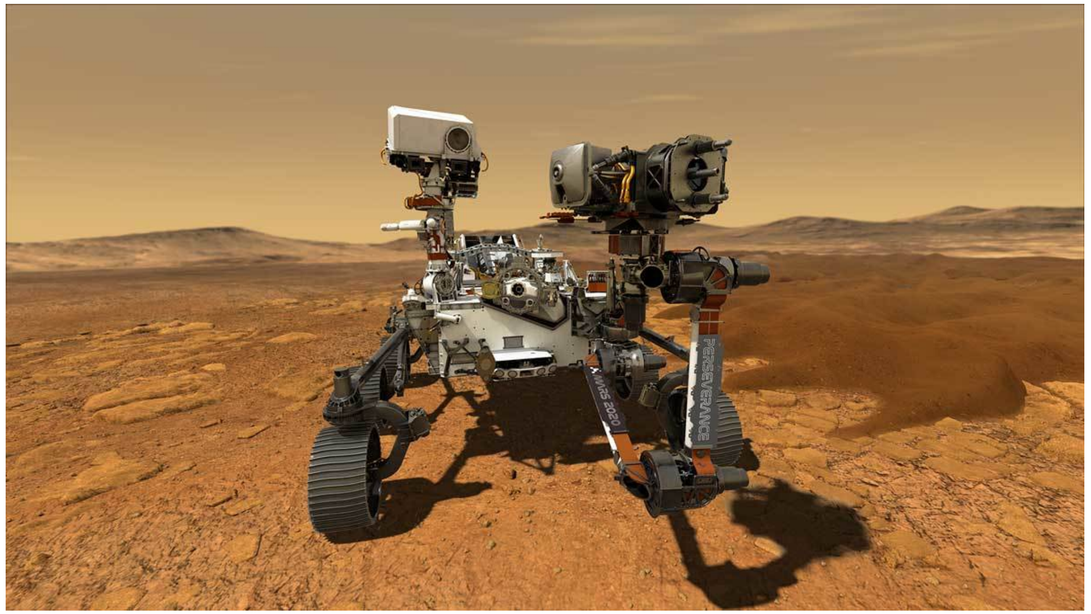
cgjohann@iastate.edu

FMICS 2023


PERSEVERANCE
MARS 2020

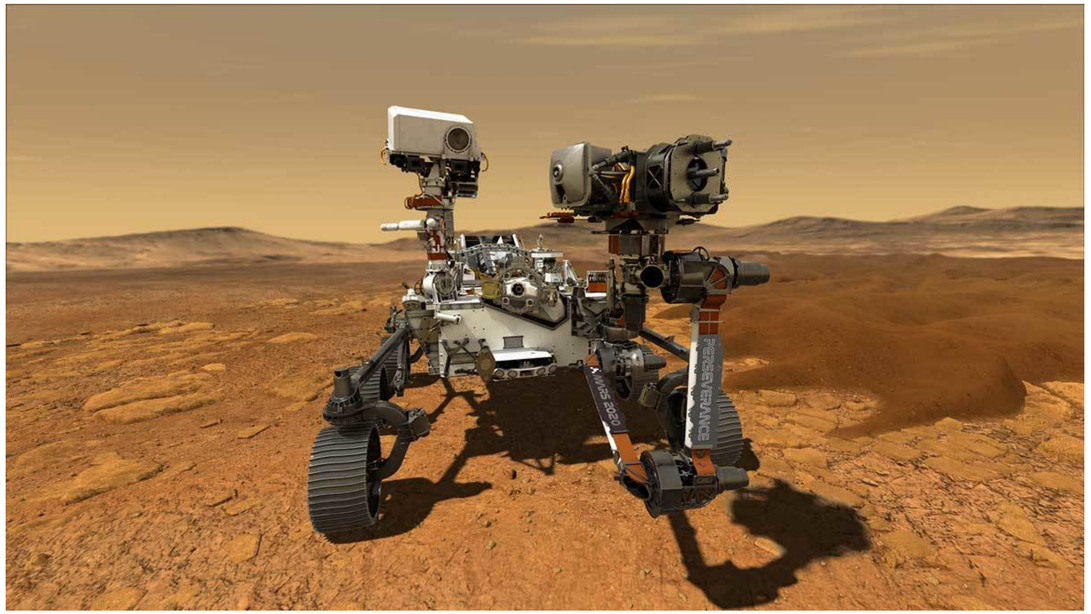


“Every file that opens is eventually closed”



“Every file that opens is eventually closed”

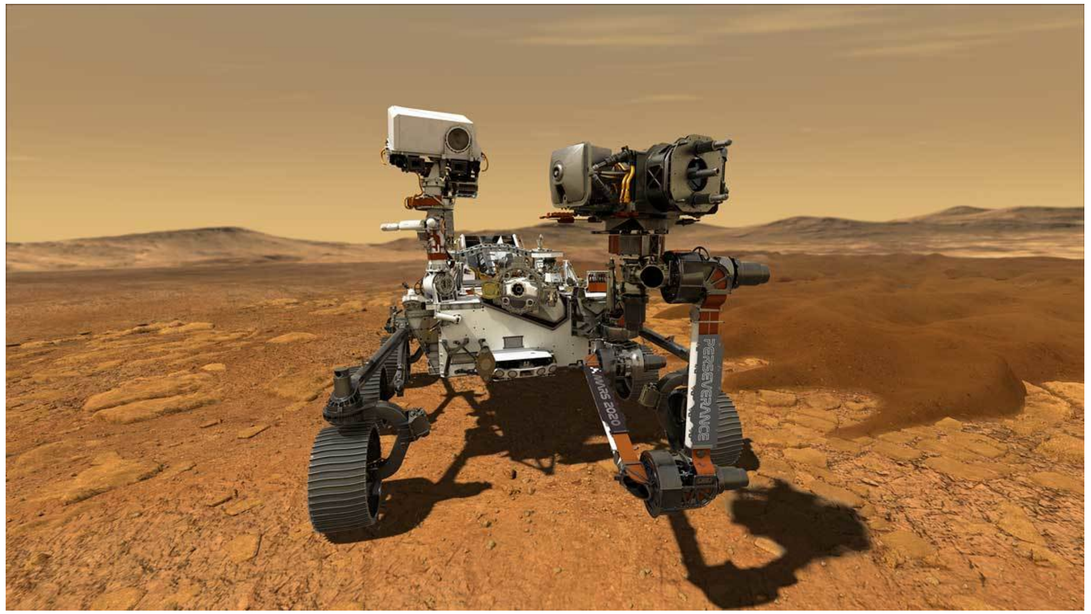
$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$



“Every file that opens is eventually closed”

$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

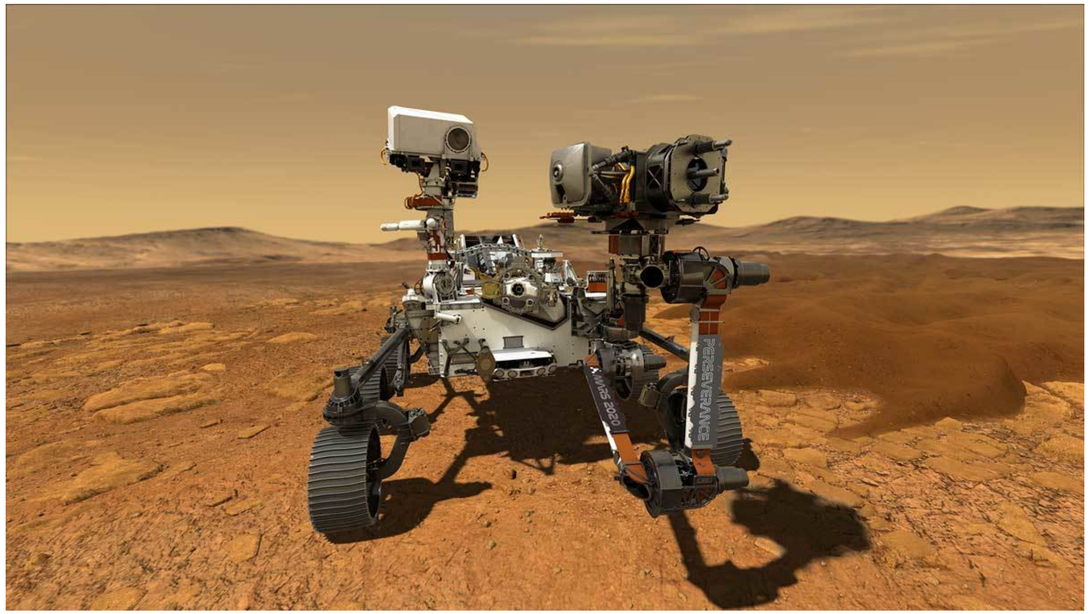


“Every file that opens is eventually closed”

$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$

1

INTRACTABLE!



“Every file that opens is eventually closed”

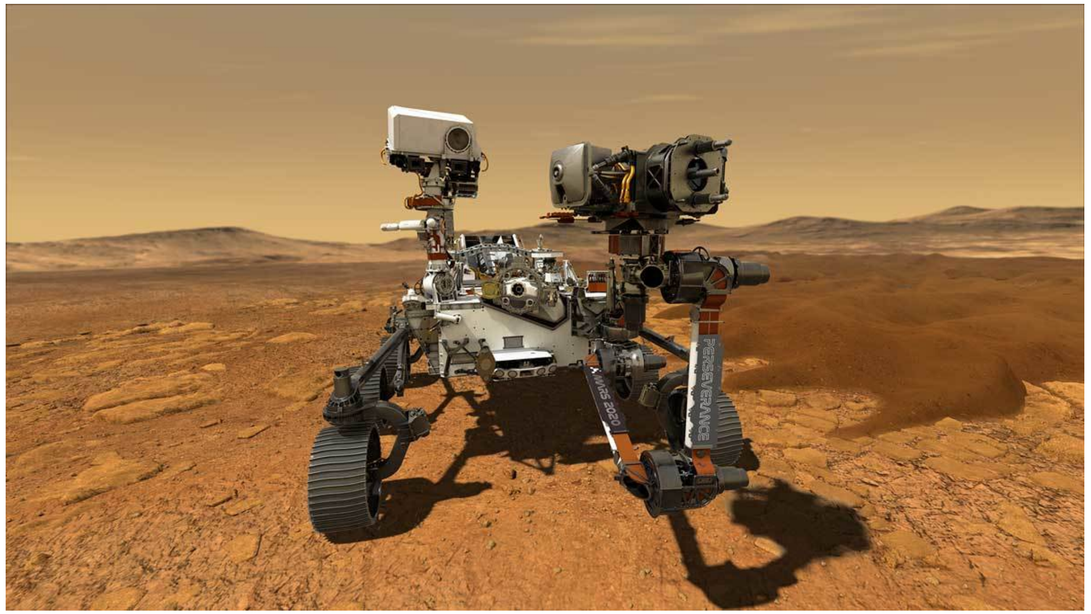
2

$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$

1

INTRACTABLE!

7



“Every file that opens is eventually closed”

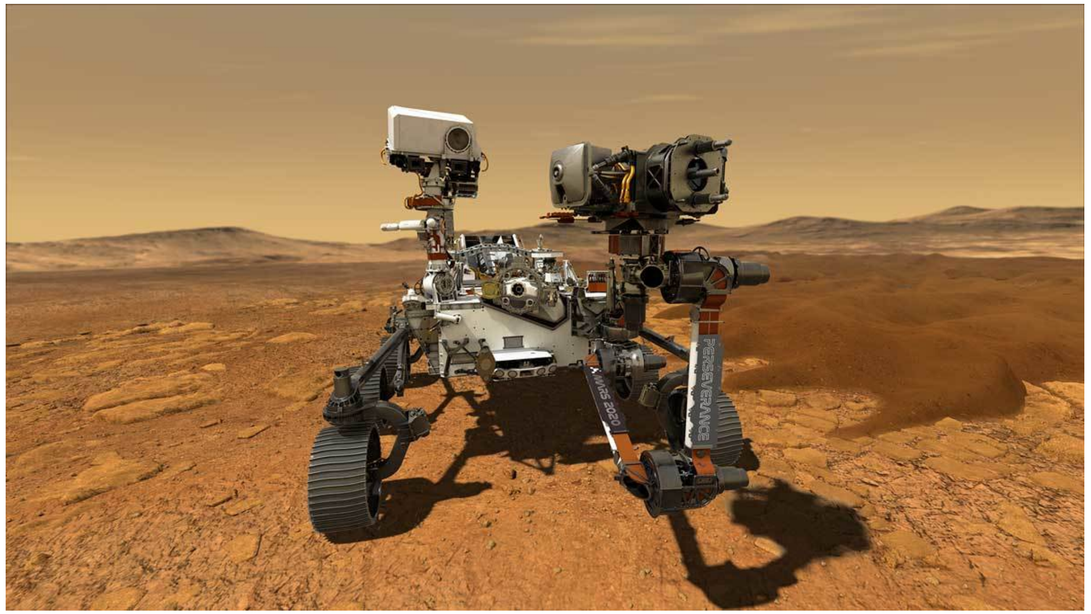
3 2

$$\forall f,\ open(f) \rightarrow \Diamond(close(f))$$

1

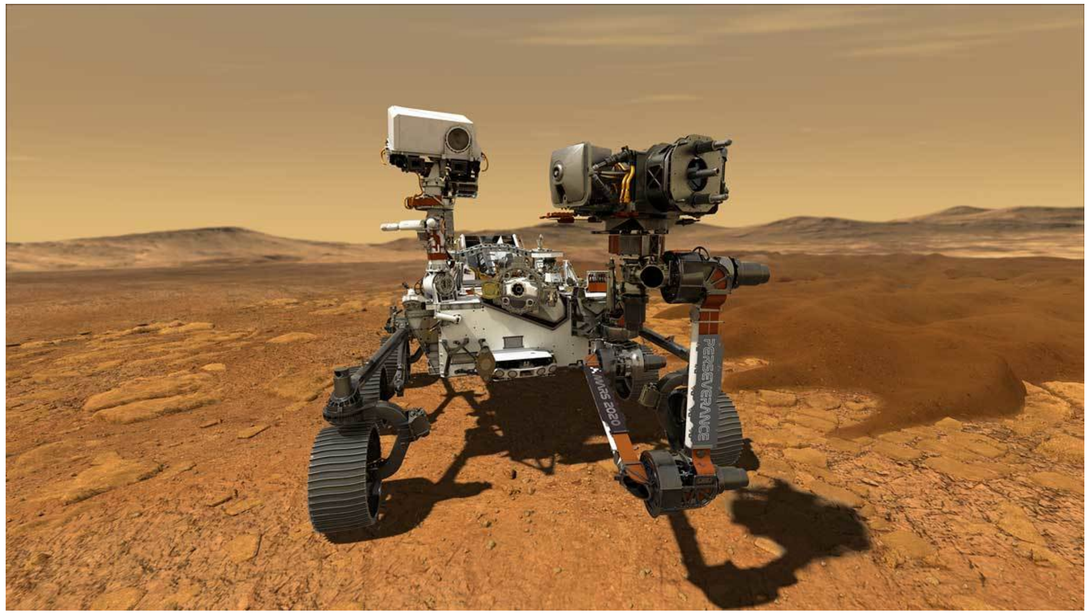
INTRACTABLE!

8



“At least k files that open are eventually closed”

$$\exists^{\geq k} f \,.\, open(f) \rightarrow \Diamond(close(f))$$

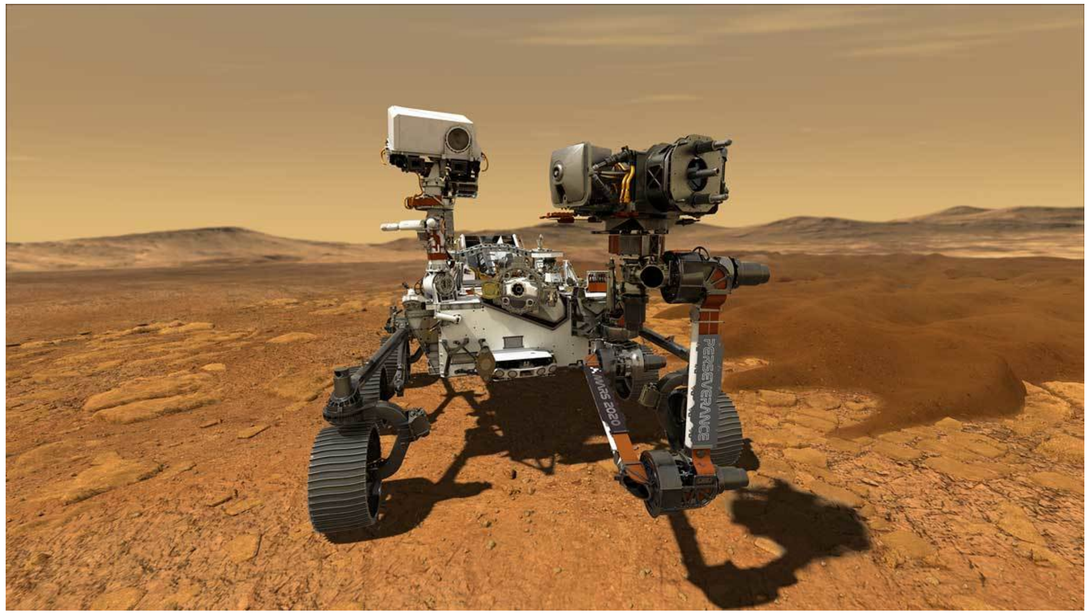


“At least k files that open are eventually closed”

$$\exists^{\geq k} f \,.\, open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

10



“At least k files that open are eventually closed”

4

$$\exists^{\geq k} f \,.\, open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

11

"Every file that opens is eventually closed"

3 2 1

$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$

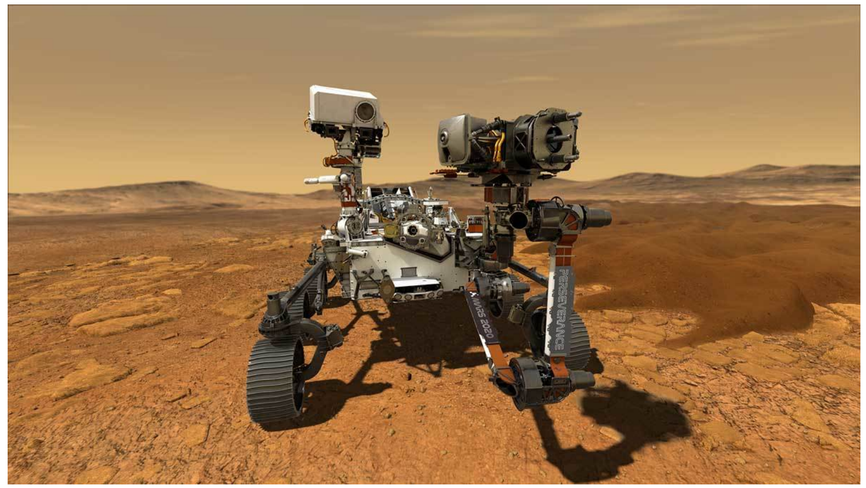
INTRACTABLE!

"At least k files that open are eventually closed"

4

$$\exists^{\geq k} f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!



12

"Every file that opens is eventually closed"

3 2 1

$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

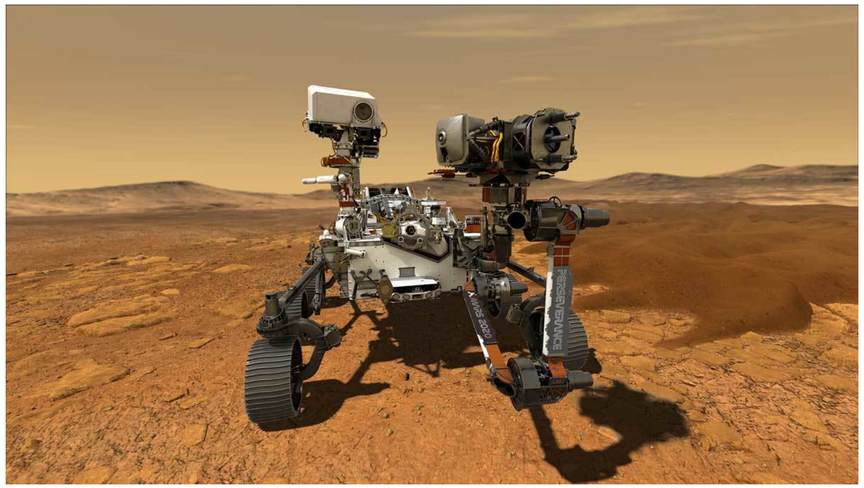
"At least k files that open are eventually closed"

4

$$\exists^{\geq k} f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

FO-LTL



13

"Every file that opens is eventually closed"

3 2 1

$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

"At least k files that open are eventually closed"

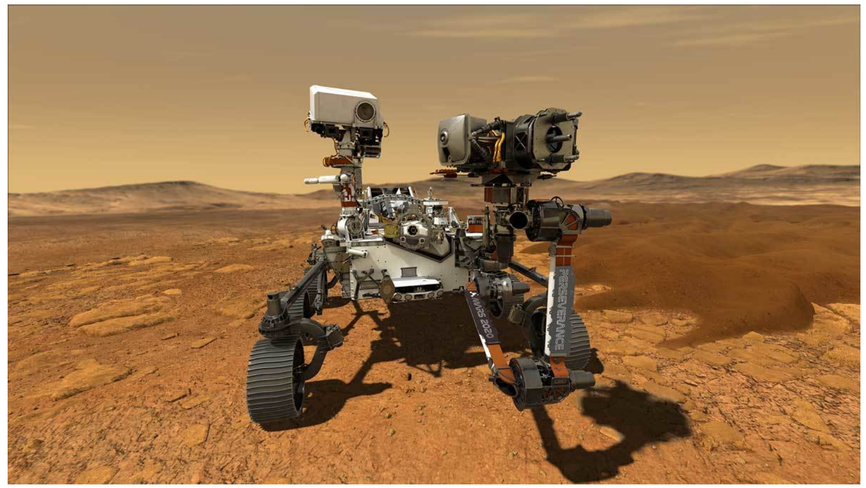
4

$$\exists^{\geq k} f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

FO-LTL

FQLTL



14

"Every file that opens is eventually closed"

3 2 1

$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

"At least k files that open are eventually closed"

4

$$\exists^{\geq k} f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

FO-LTL

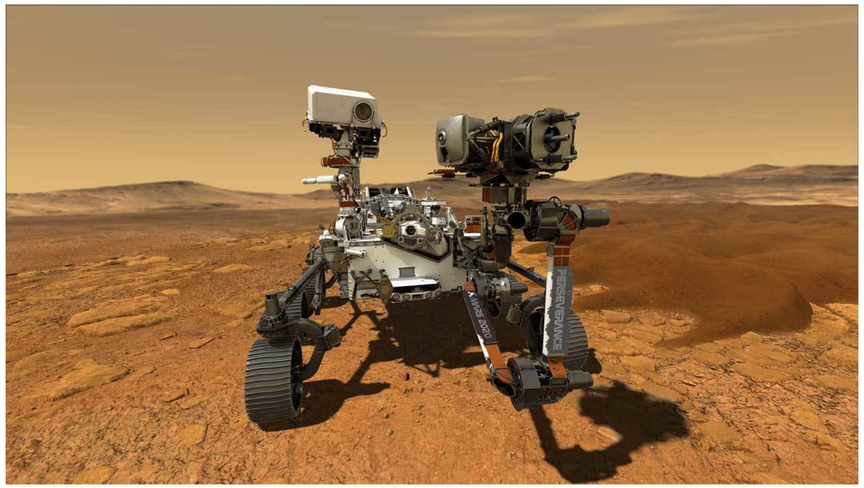
FQLTL

QPTL

QLTL

VLTL

FO-LTL^{fin}



15

"Every file that opens is eventually closed"

3 2 1

$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

"At least k files that open are eventually closed"

4

$$\exists^{\geq k} f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

~~FO-LTL~~

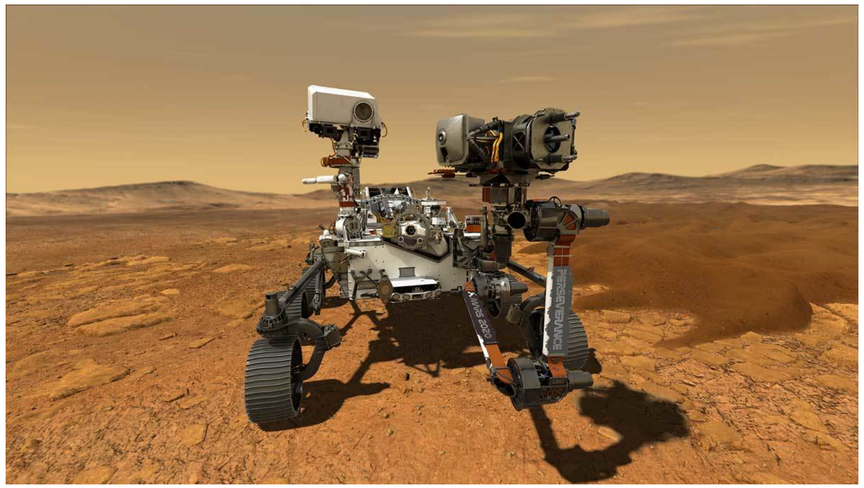
~~FQLTL~~

~~QPTL~~

~~QLTL~~

~~VLTL~~

~~FO-LTL^{fin}~~



16

"Every file that opens is eventually closed"

3 2 1

$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

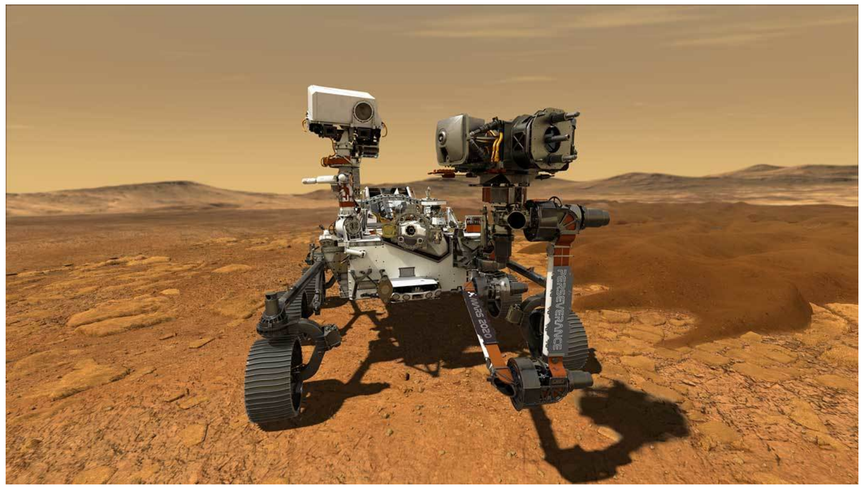
"At least k files that open are eventually closed"

4

$$\exists^{\geq k} f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

- Bounded mission time



17

"Every file that opens is eventually closed"

3 2 1

$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

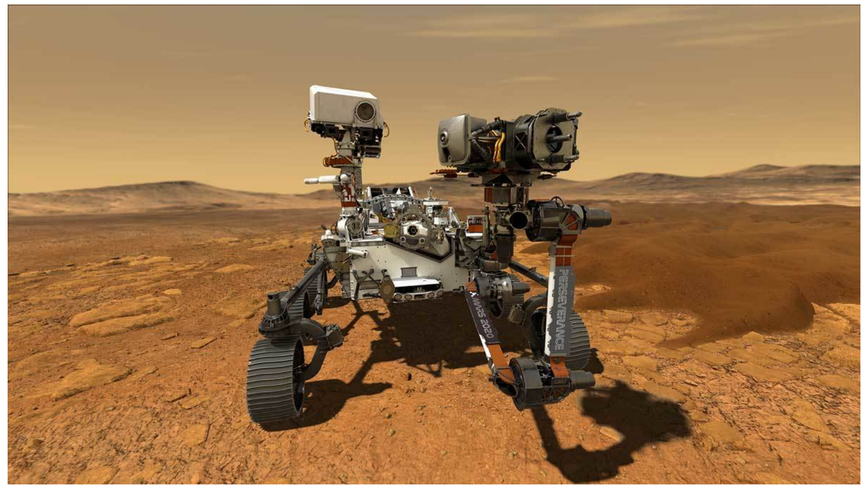
"At least k files that open are eventually closed"

4

$$\exists^{\geq k} f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE!

- Bounded mission time
- OS Limits: Max num. files open at once



18

"Every file that opens is eventually closed"

3 2 1

$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE?

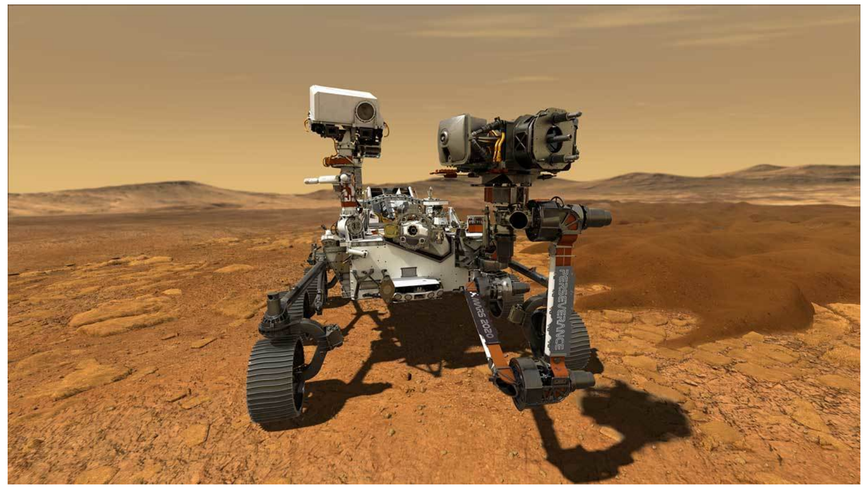
"At least k files that open are eventually closed"

4

$$\exists^{\geq k} f\,.\,open(f) \rightarrow \Diamond(close(f))$$

INTRACTABLE?

- **Bounded mission time**
- **OS Limits: Max num. files open at once**



19

“Every file that opens is eventually closed”

$$\forall f\,.\,open(f) \rightarrow \Diamond(close(f))$$

1

20

“Every file that opens is eventually closed”

$$\forall f\,.\,open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

21

“Every file that opens is eventually closed”

$$\forall f\,.\,open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

We use **MLTL** (Mission-time Linear Temporal Logic) — an ***integer-bounded*** variant of LTL over ***finite traces.***

22

“Every file that opens is eventually closed”

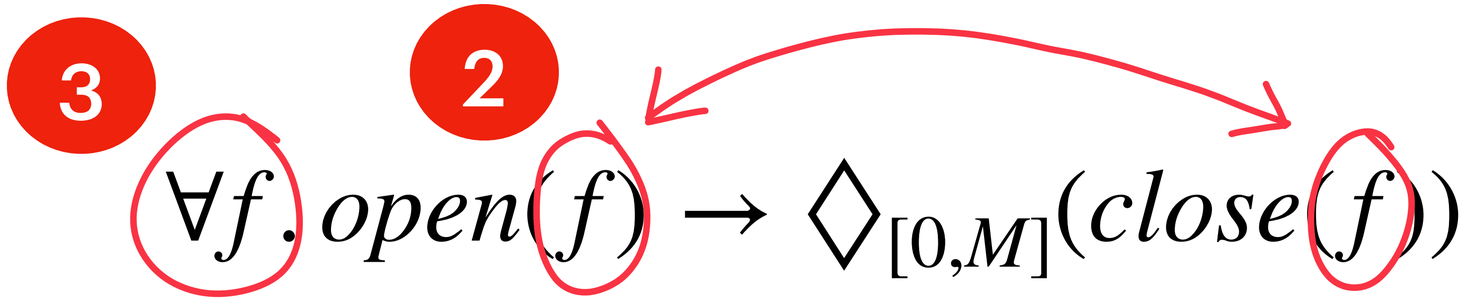
$$\forall f\,.\,open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

We use **MLTL** (Mission-time Linear Temporal Logic) — an ***integer-bounded*** variant of LTL over ***finite traces.***

MLTL monitors provide ***tight memory and computation time bounds.***

23

“Every file that opens is eventually closed”

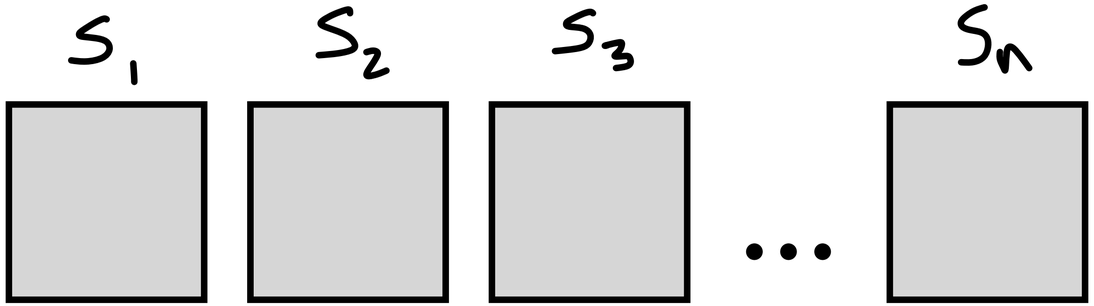


24

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

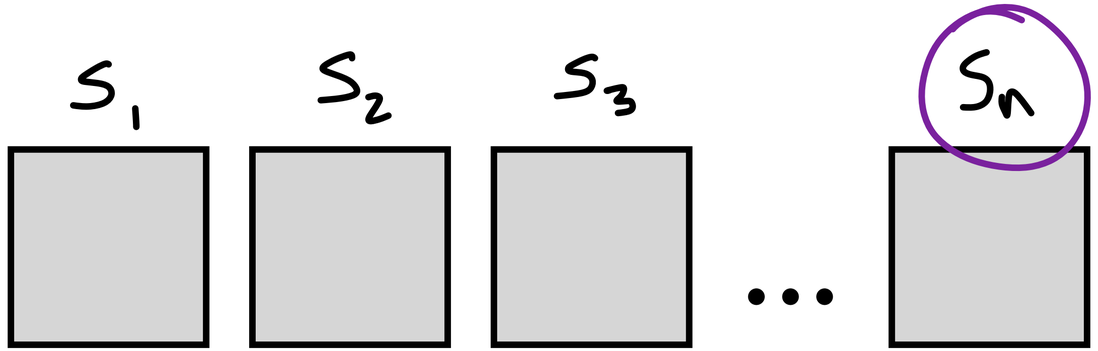


25

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$



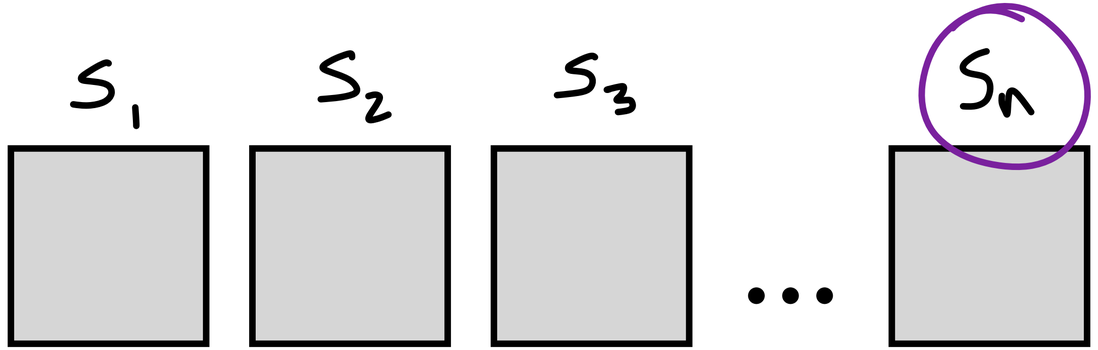
Only n files can be open at once

26

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$



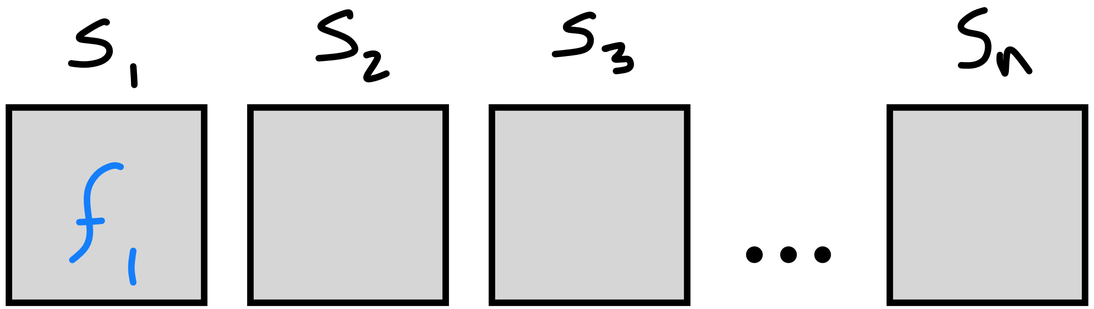
Only n files can be open at once

Default for Linux: 1024

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

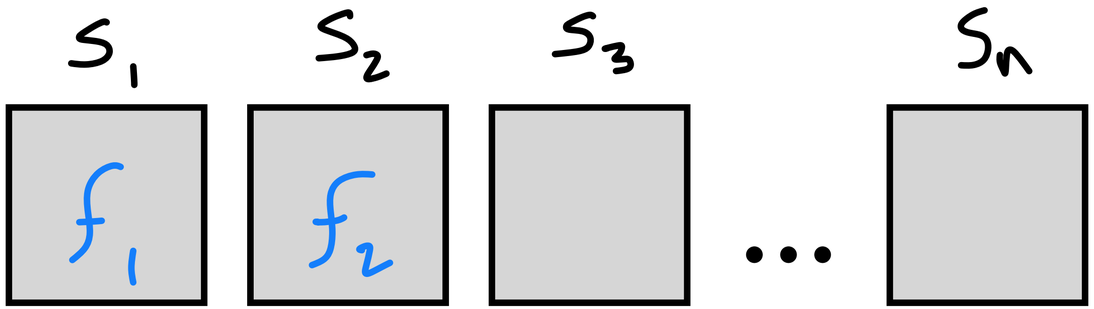


28

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

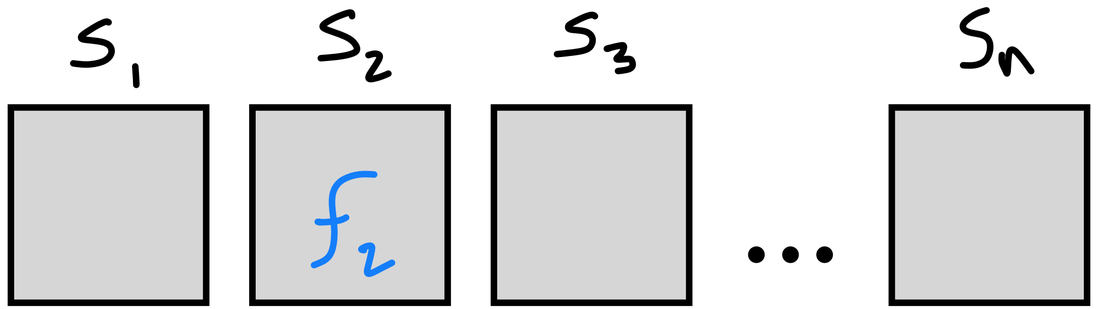


29

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

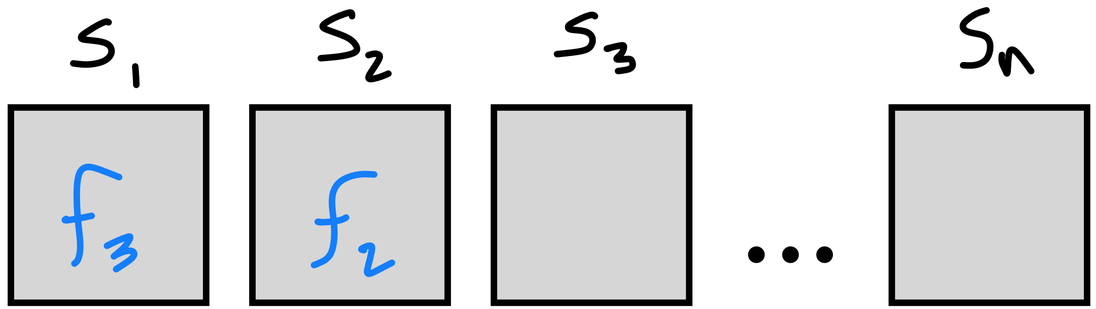


30

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$



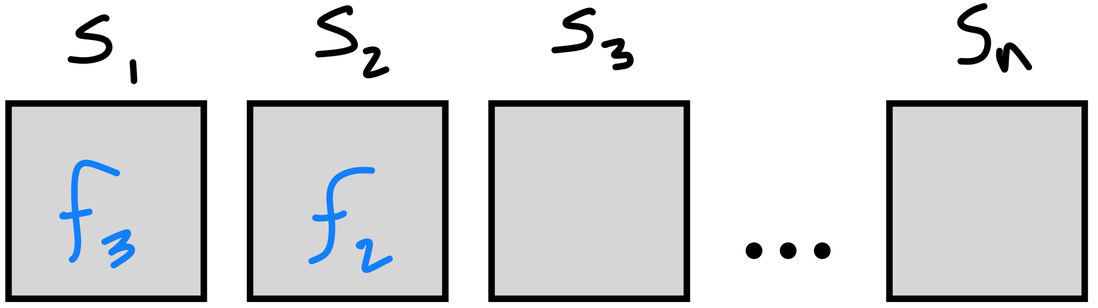
31

"Every file that opens is eventually closed"

3 2

$$\forall f.\ open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

$$open_{S_1} = open(f_3)$$
$$close_{S_1} = close(f_3)$$



32

"Every file that opens is eventually closed"

3 2

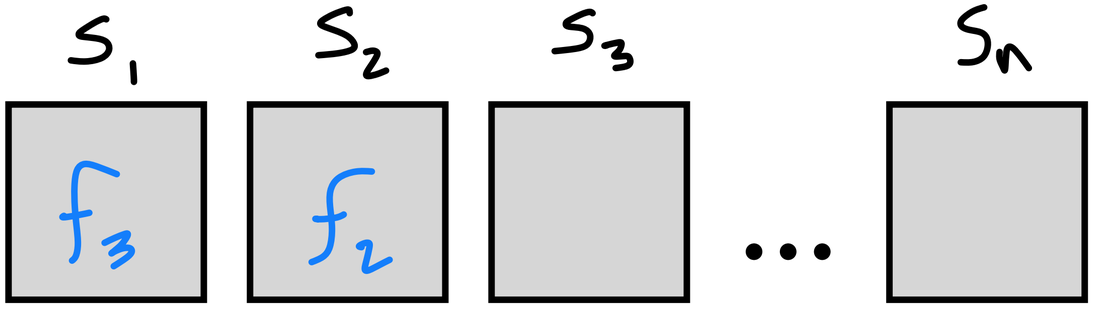
$$\forall f.\ open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

$open_{S_1} = open(f_3)$

$close_{S_1} = close(f_3)$

$open_{S_2} = open(f_2)$

$close_{S_2} = close(f_2)$



33

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

$open_{S_1} = open(f_3)$

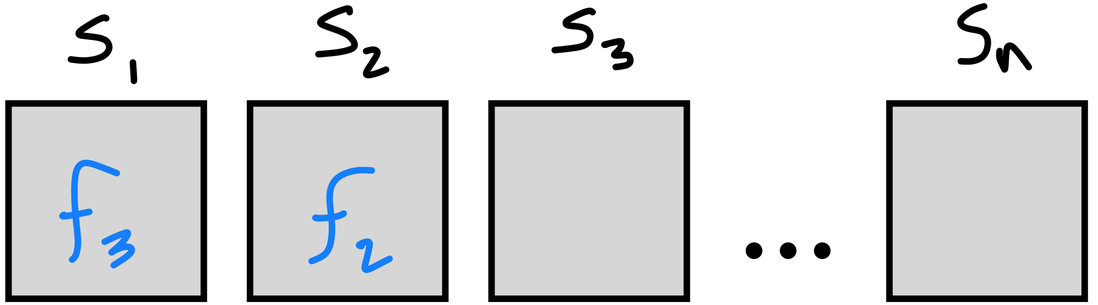
$close_{S_1} = close(f_3)$

$open_{S_2} = open(f_2)$

$close_{S_2} = close(f_2)$

$open_{S_3} = false$

$close_{S_3} = false$



34

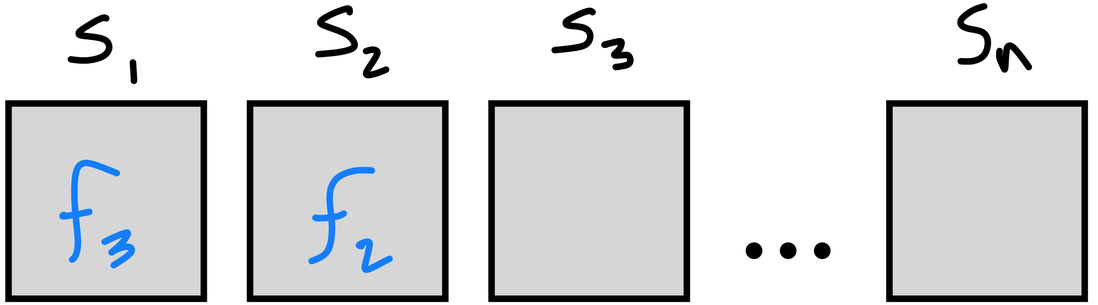
"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

Atomic Propositions

$$open_{S_1} = open(f_3)$$
$$close_{S_1} = close(f_3)$$
$$open_{S_2} = open(f_2)$$
$$close_{S_2} = close(f_2)$$
$$open_{S_3} = false$$
$$close_{S_3} = false$$



35

“Every file that opens is eventually closed”

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

$$open_{S_1} \rightarrow \Diamond_{[0,M]}(close_{S_1}) \wedge$$

$$open_{S_2} \rightarrow \Diamond_{[0,M]}(close_{S_2}) \wedge$$

$$\ldots$$

$$open_{S_n} \rightarrow \Diamond_{[0,M]}(close_{S_n})$$

36

"Every file that opens is eventually closed"

3 2

$$\forall f.\ open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

$open_{S_2}$	$close_{S_2}$	S_2
true	*false*	f_2

$$open_{S_2} \rightarrow \Diamond_{[0,M]}(close_{S_2})$$

37

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

$$open_{S_2} \rightarrow \Diamond_{[0,M]}(close_{S_2})$$

$open_{S_2}$	$close_{S_2}$	S_2
true	*false*	f_2
true	*false*	f_4

38

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

$$open_{S_2} \rightarrow \Diamond_{[0,M]}(close_{S_2})$$

$open_{S_2}$	$close_{S_2}$	S_2
true	*false*	f_2
true	*false*	f_4
false	*true*	f_4

39

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

Incorrect behavior, but monitor won't catch it!

$$open_{S_2} \rightarrow \Diamond_{[0,M]}(close_{S_2})$$

$open_{S_2}$	$close_{S_2}$	S_2
true	*false*	f_2
true	*false*	f_4
false	*true*	f_4

40

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

"no change"

$$open_{S_2} \rightarrow \Diamond_{[0,M]}(close_{S_2})$$

nc_{S_2}	$open_{S_2}$	$close_{S_2}$	S_2
true	*true*	*false*	f_2
false	*true*	*false*	f_4
true	*false*	*true*	f_4

41

"Every file that opens is eventually closed"

3 2

$$\forall f.\ open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

"no change"

$$open_{S_2} \rightarrow \Diamond_{[0,M]}(close_{S_2})$$

nc_{S_2}	$open_{S_2}$	$close_{S_2}$	S_2
true	*true*	*false*	f_2
false	*true*	*false*	f_4
true	*false*	*true*	f_4

42

"Every file that opens is eventually closed"

3 2

$$\forall f.\ open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

"no change"

$$open_{S_2} \rightarrow \Diamond_{[0,M]}(close_{S_2})$$

$$open_{S_2} \rightarrow ((nc_2)\mathcal{U}_{[0,M]}(nc_2 \wedge close_{S_2}))$$

nc_{S_2}	$open_{S_2}$	$close_{S_2}$	S_2
true	*true*	*false*	f_2
false	*true*	*false*	f_4
true	*false*	*true*	f_4

43

“Every file that opens is eventually closed”

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

$$open_{S_1} \rightarrow \Diamond_{[0,M]}(close_{S_1}) \wedge$$

$$open_{S_2} \rightarrow \Diamond_{[0,M]}(close_{S_2}) \wedge$$

$$\ldots$$

$$open_{S_n} \rightarrow \Diamond_{[0,M]}(close_{S_n})$$

44

"Every file that opens is eventually closed"

3 2

$$\forall f.\, open(f) \rightarrow \Diamond_{[0,M]}(close(f))$$

$$open_{S_1} \rightarrow \Diamond_{[0,M]}(close_{S_1}) \wedge$$
$$open_{S_2} \rightarrow \Diamond_{[0,M]}(close_{S_2}) \wedge$$
$$\ldots$$
$$open_{S_n} \rightarrow \Diamond_{[0,M]}(close_{S_n})$$

$\Rightarrow$

$$open_{S_1} \rightarrow ((nc_1)\mathcal{U}_{[0,M]}(nc_1 \wedge close_{S_1})) \wedge$$
$$open_{S_2} \rightarrow ((nc_2)\mathcal{U}_{[0,M]}(nc_2 \wedge close_{S_2})) \wedge$$
$$\ldots$$
$$open_{S_n} \rightarrow ((nc_n)\mathcal{U}_{[0,M]}(nc_n \wedge close_{S_n}))$$

45

“Every file that opens is eventually closed”

$$open_{S_1} \rightarrow ((nc_1)\mathcal{U}_{[0,M]}(nc_1 \wedge close_{S_1})) \wedge$$
$$open_{S_2} \rightarrow ((nc_2)\mathcal{U}_{[0,M]}(nc_2 \wedge close_{S_2})) \wedge$$
$$\ldots$$
$$open_{S_n} \rightarrow ((nc_n)\mathcal{U}_{[0,M]}(nc_n \wedge close_{S_n}))$$

46

"Every file that opens is eventually closed"

$$open_{S_1} \rightarrow ((nc_1)\mathcal{U}_{[0,M]}(nc_1 \wedge close_{S_1})) \wedge$$
$$open_{S_2} \rightarrow ((nc_2)\mathcal{U}_{[0,M]}(nc_2 \wedge close_{S_2})) \wedge$$
$$\ldots$$
$$open_{S_n} \rightarrow ((nc_n)\mathcal{U}_{[0,M]}(nc_n \wedge close_{S_n}))$$

100% Pure MLTL

47

"Every file that opens is eventually closed"

$$open_{S_1} \rightarrow ((nc_1)\mathcal{U}_{[0,M]}(nc_1 \wedge close_{S_1})) \wedge$$
$$open_{S_2} \rightarrow ((nc_2)\mathcal{U}_{[0,M]}(nc_2 \wedge close_{S_2})) \wedge$$
$$\ldots$$
$$open_{S_n} \rightarrow ((nc_n)\mathcal{U}_{[0,M]}(nc_n \wedge close_{S_n}))$$

100% Pure MLTL

(Not quantifier elimination...)

48

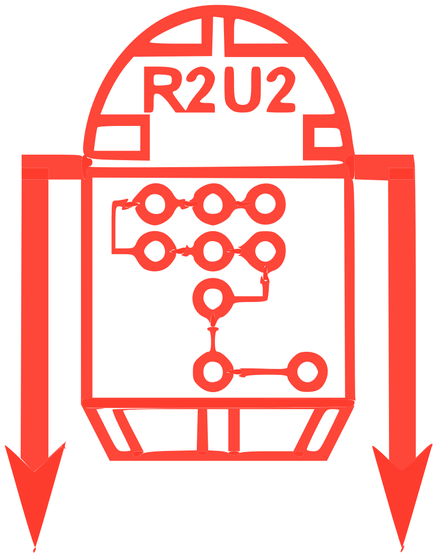
“Every file that opens is eventually closed”

```
STRUCT
  File: { open, close: bool};

INPUT
  open1, close1, …, openn, closen : bool;

DEFINE
  f1 := File(open1, close1);
  …
  fn := File(openn, closen);
  FSet := {f1, …, fn};

FTSPEC
  foreach(f : FSet) (f.open -> F[0,M](f.close));
```



49

“At least k files that open are eventually closed”

4

$$\exists^{\geq k} f . open(f) \rightarrow \Diamond(close(f))$$

50

Now that we can monitor these specifications with guarantees, how can we *reduce* their *encoding sizes*?

51

Now that we can monitor these specifications with guarantees, how can we *reduce* their *encoding sizes*?

Rewrite Rules!

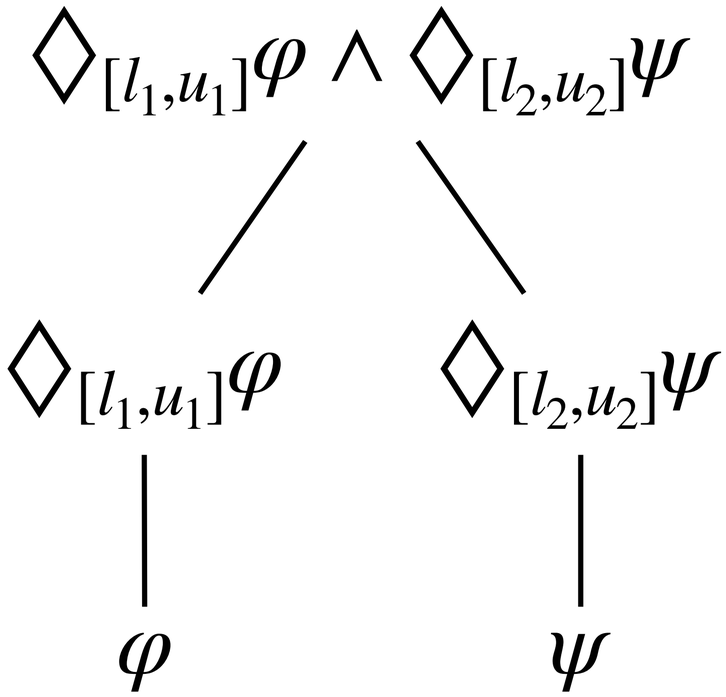
52

How much space does this require to monitor?

$$\Diamond_{[l_1,u_1]}\varphi \wedge \Diamond_{[l_2,u_2]}\psi$$

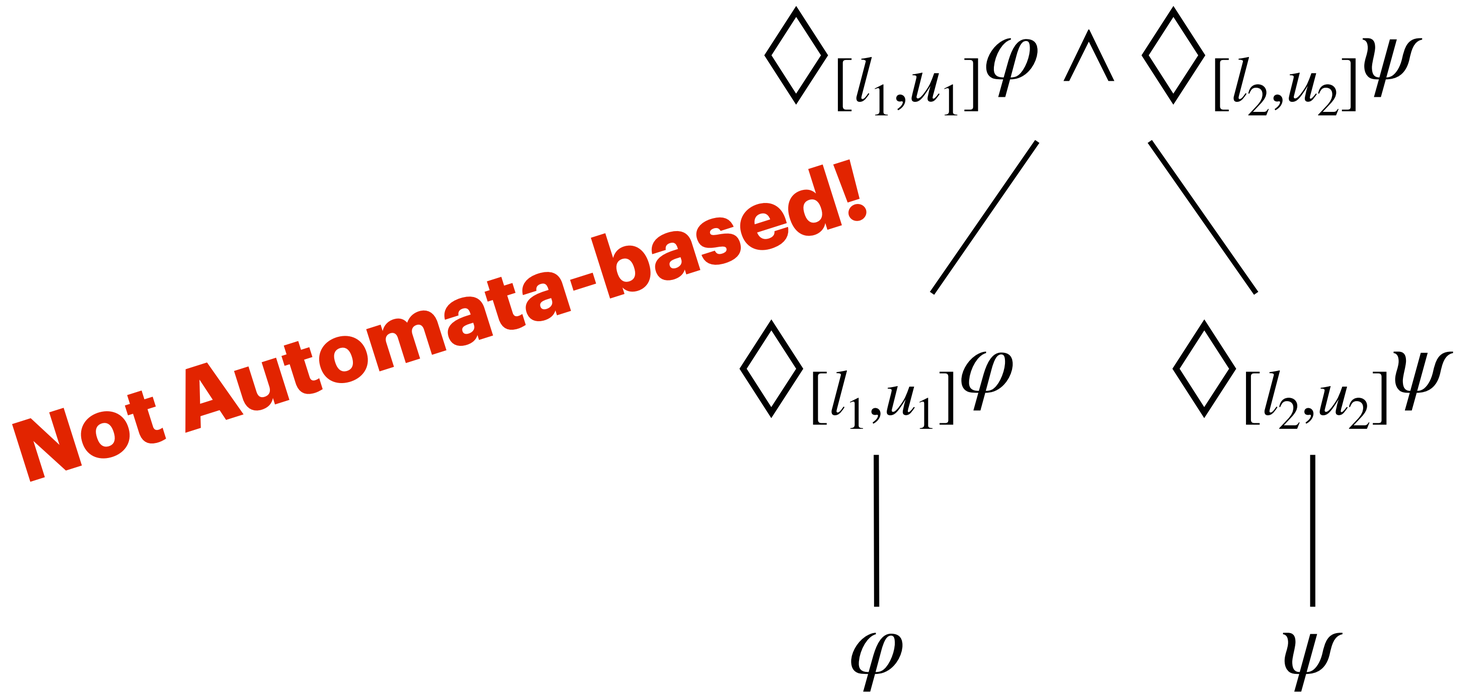
53

How much space does this require to monitor?



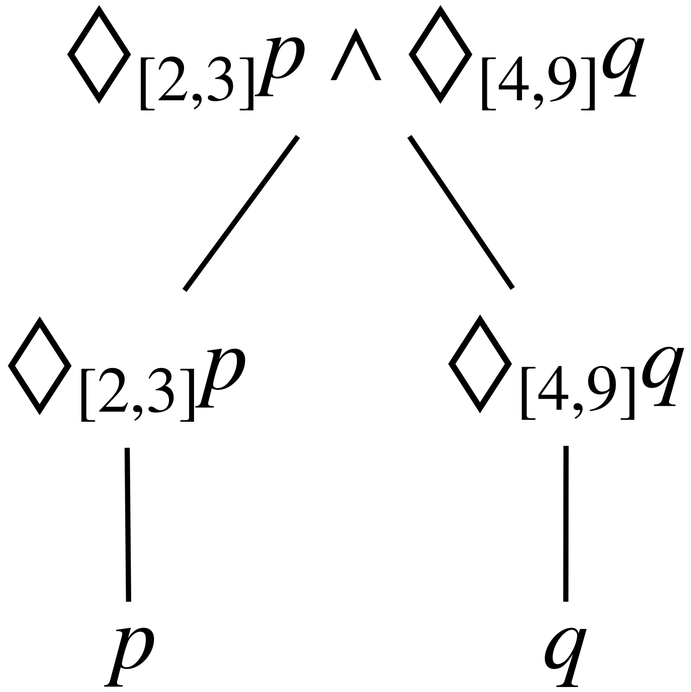
54

How much space does this require to monitor?



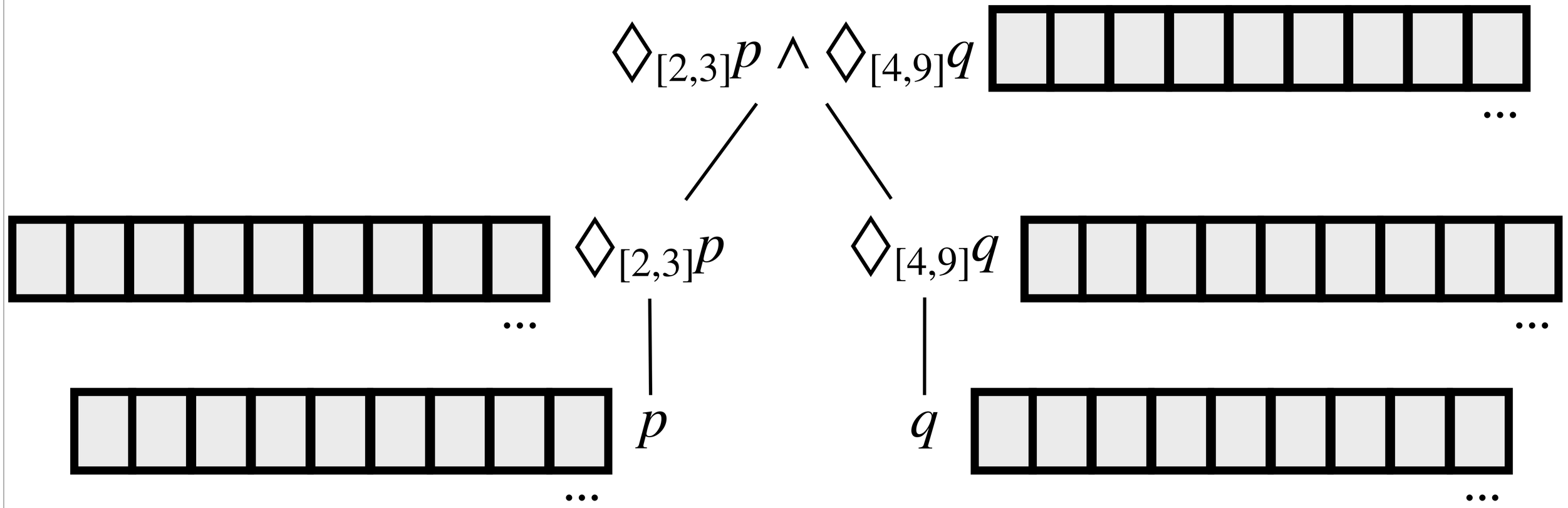
55

How much space does this require to monitor?



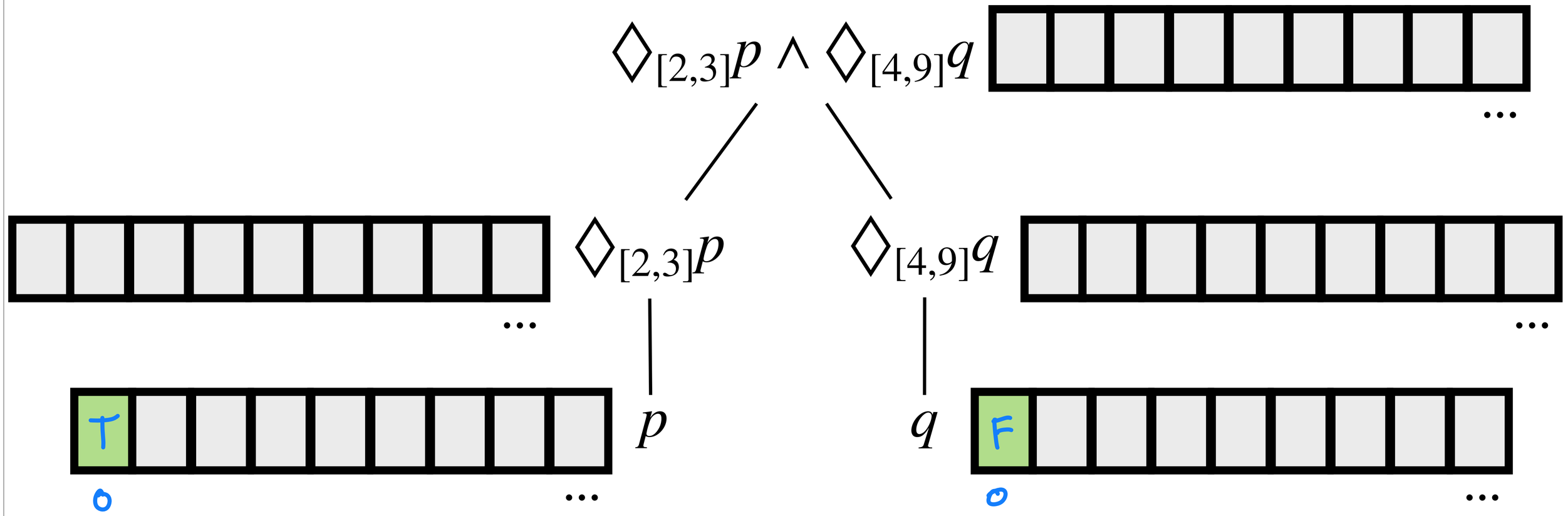
56

How much space does this require to monitor?



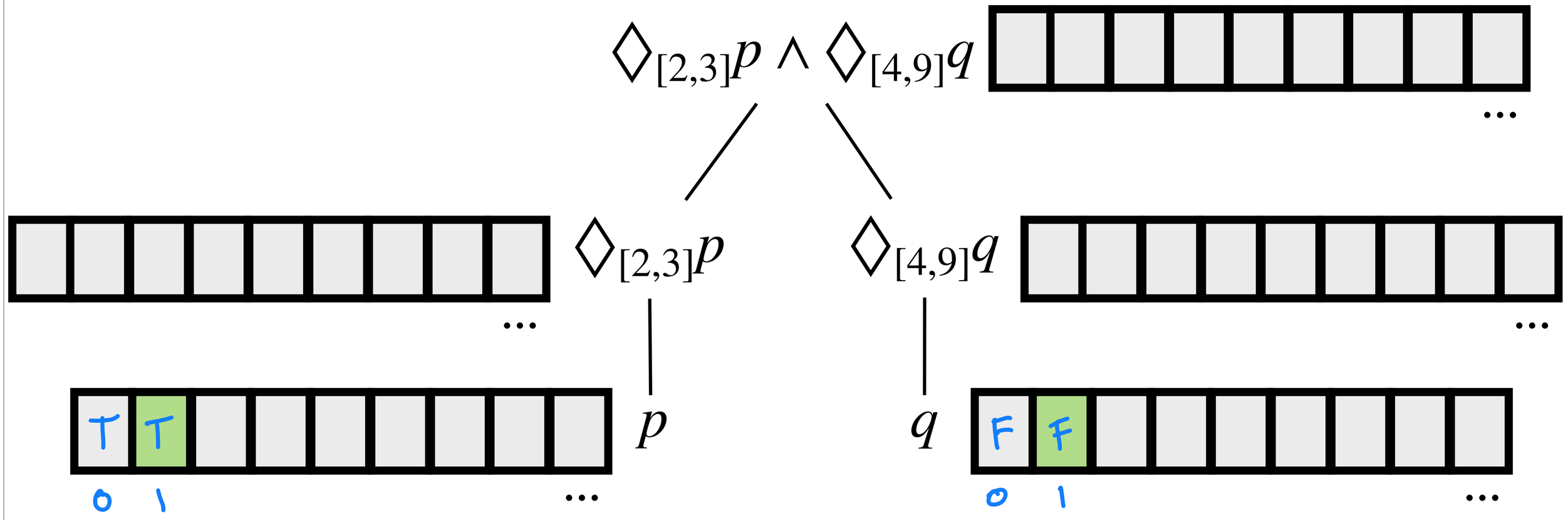
57

How much space does this require to monitor?



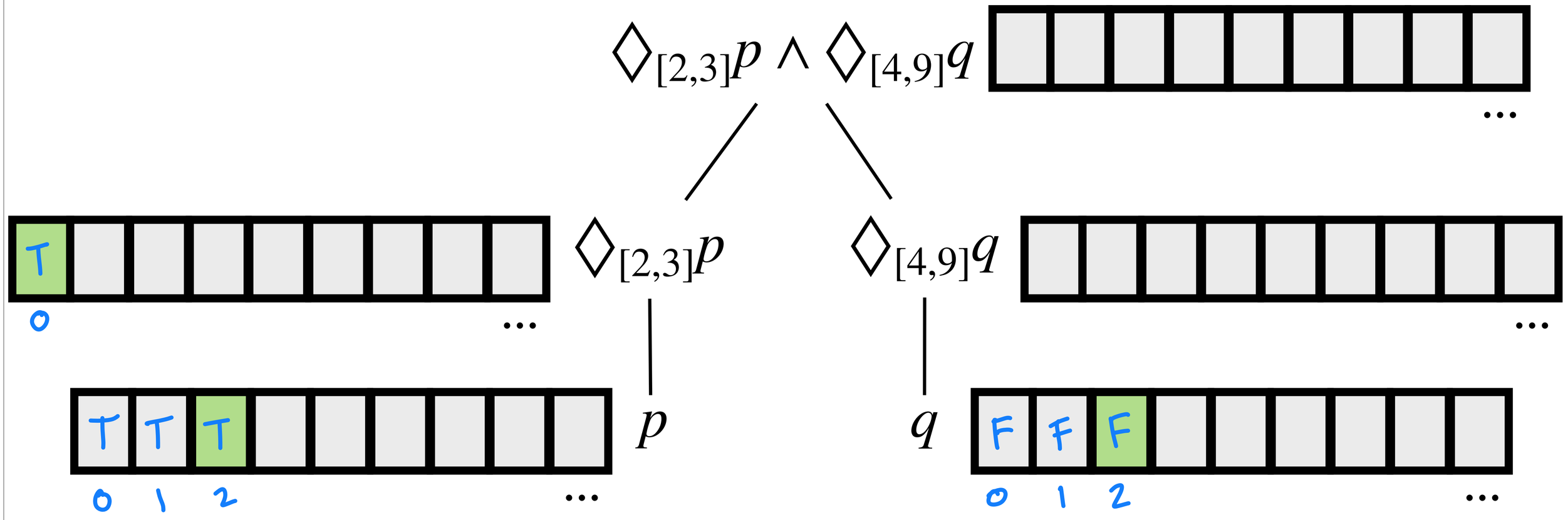
58

How much space does this require to monitor?



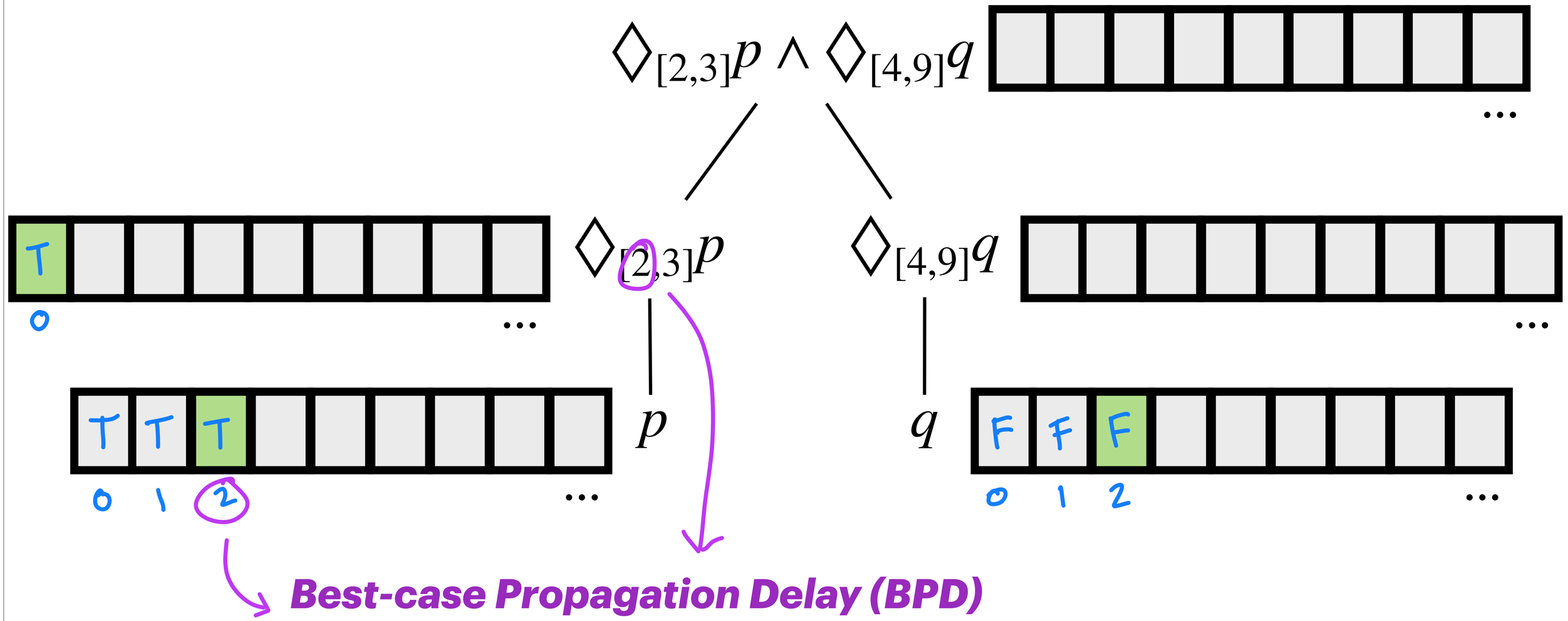
59

How much space does this require to monitor?



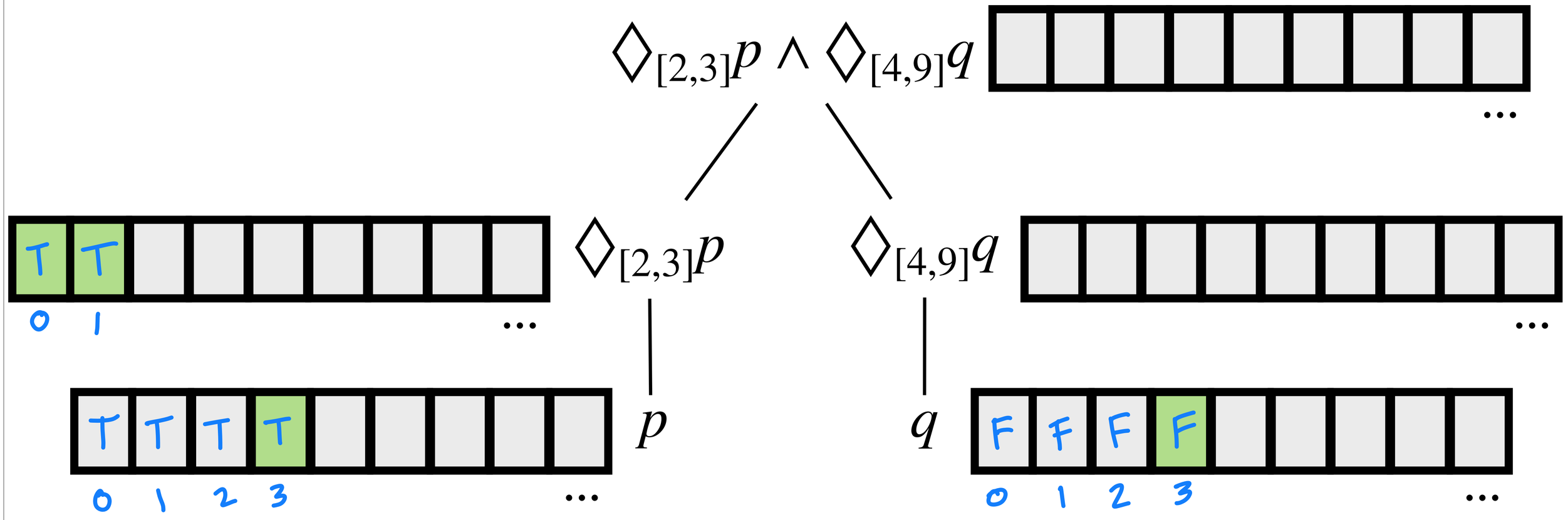
60

How much space does this require to monitor?



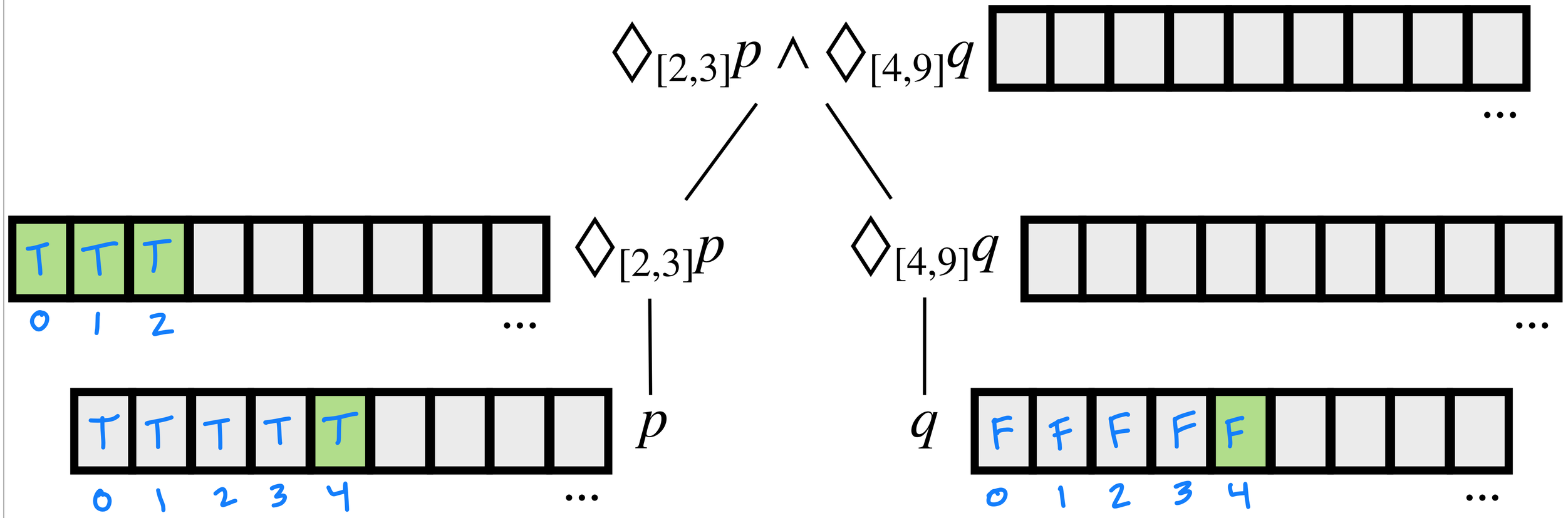
61

How much space does this require to monitor?



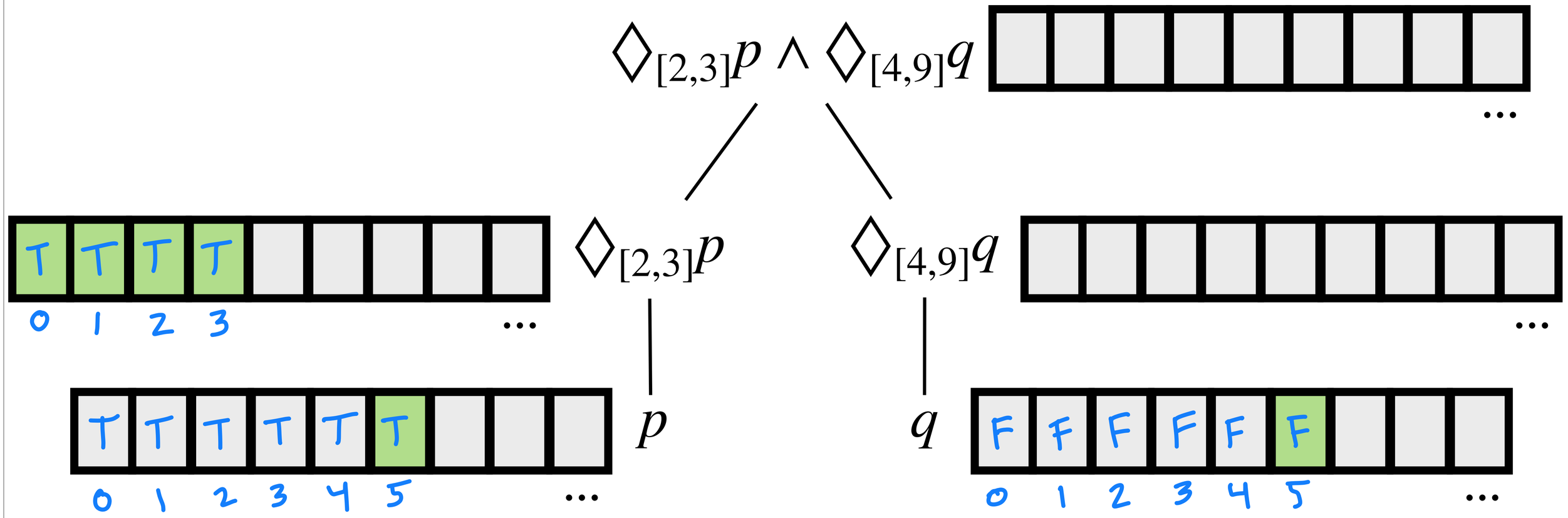
62

How much space does this require to monitor?



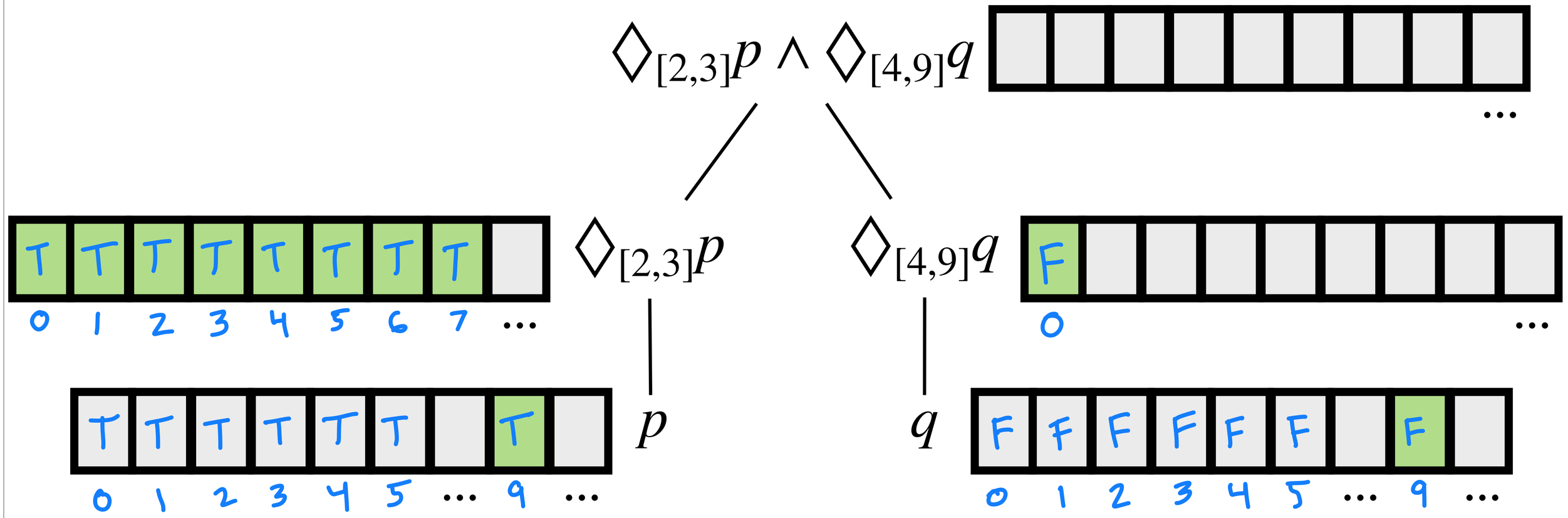
63

How much space does this require to monitor?



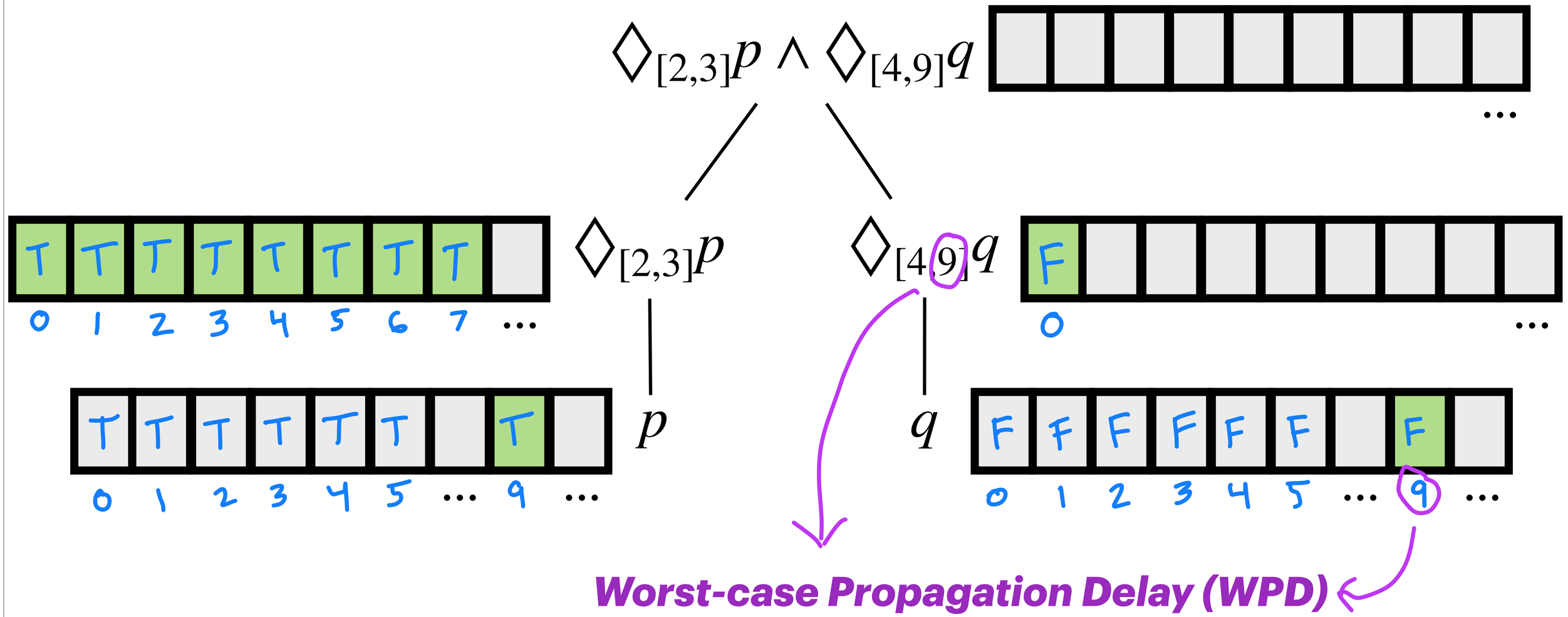
64

How much space does this require to monitor?



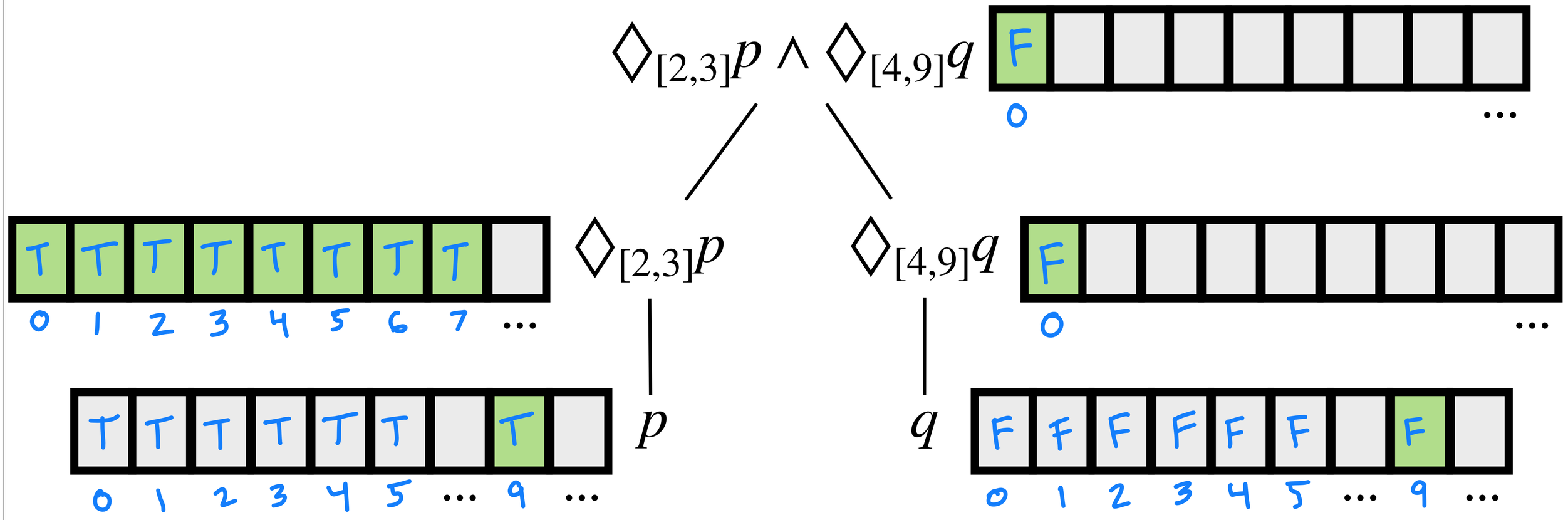
65

How much space does this require to monitor?



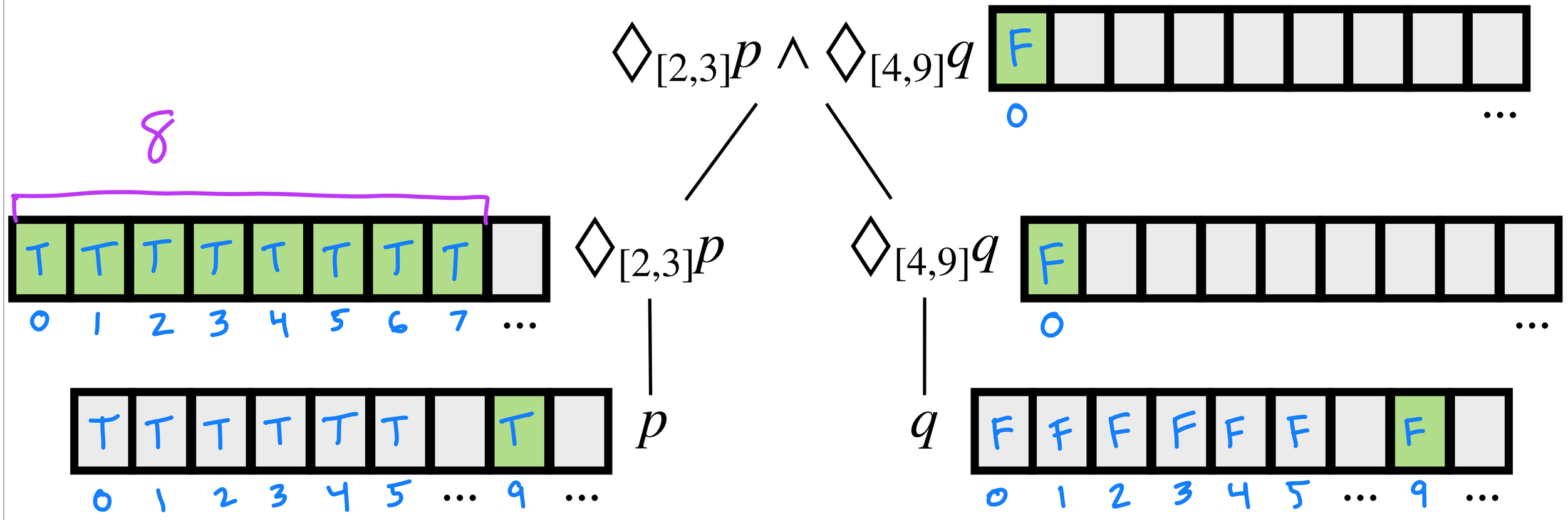
66

How much space does this require to monitor?



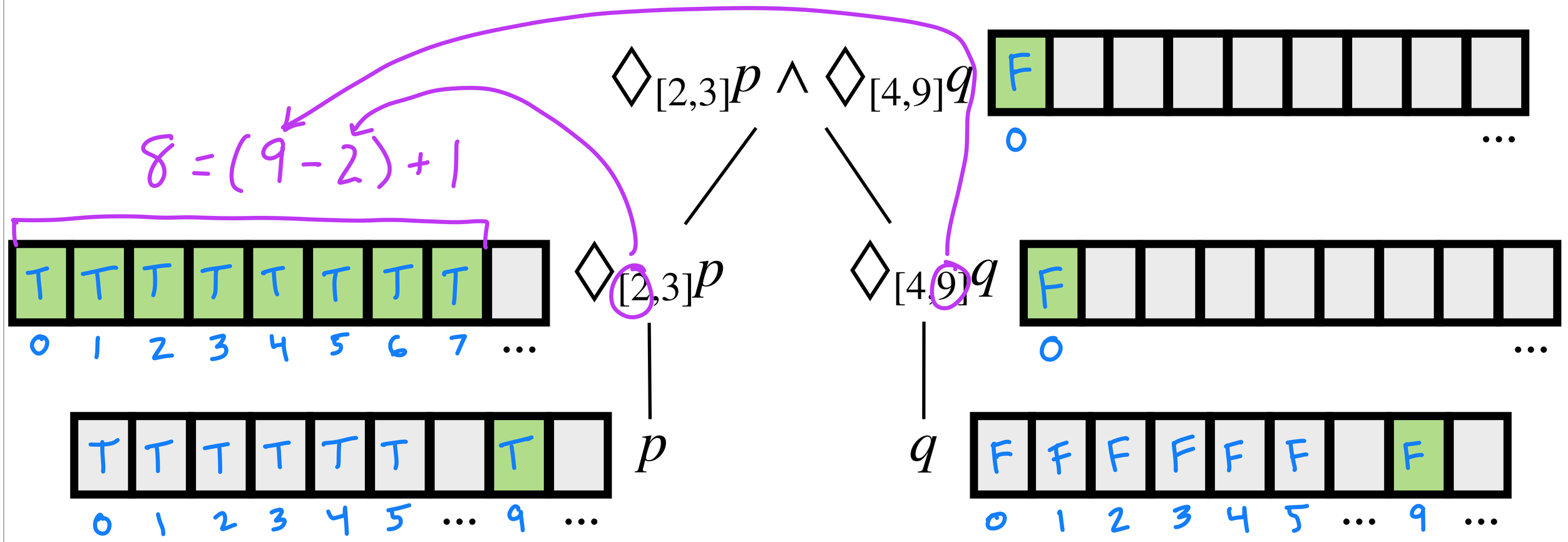
67

How much space does this require to monitor?



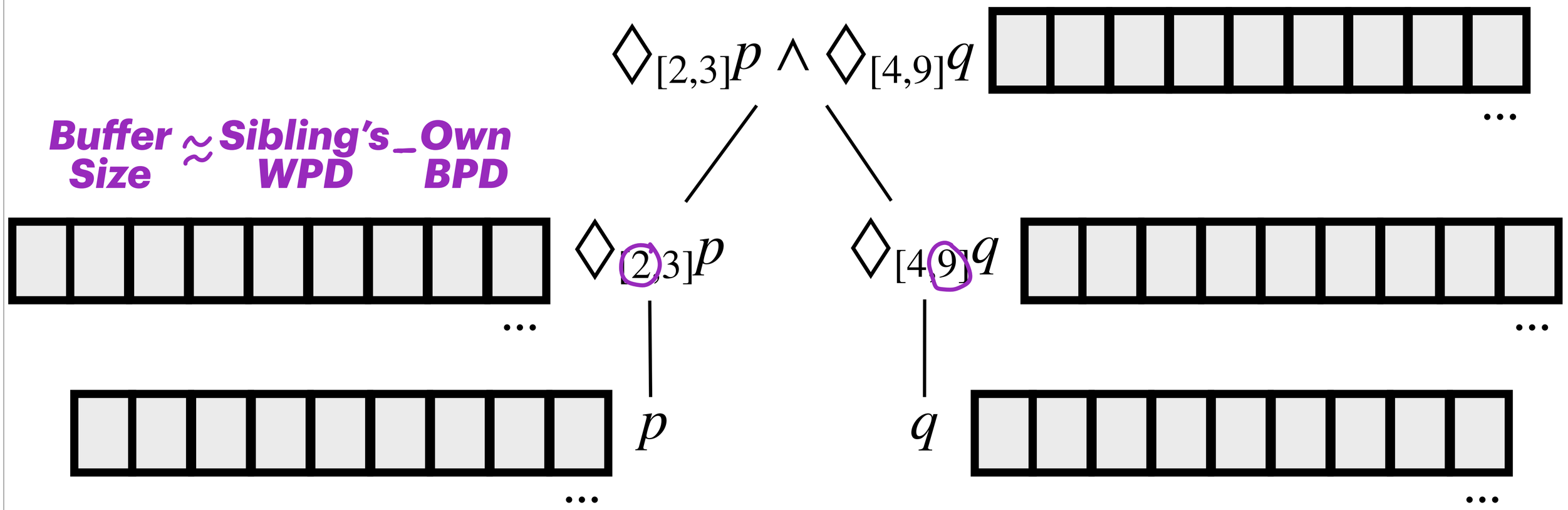
68

How much space does this require to monitor?



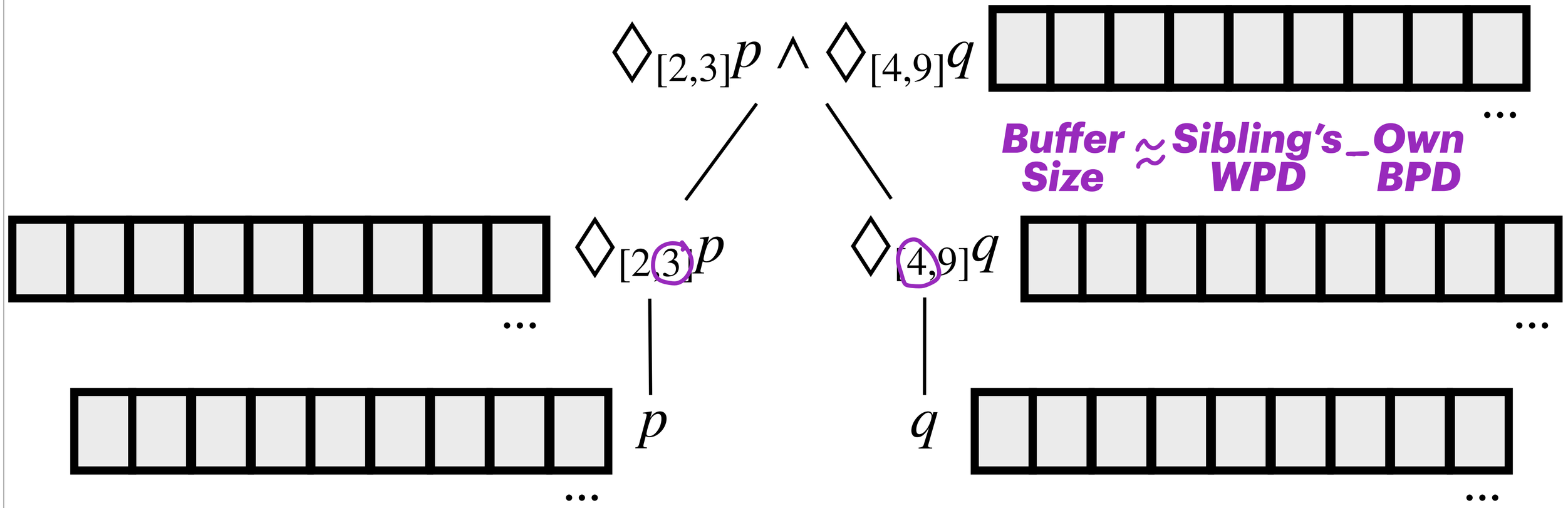
69

How much space does this require to monitor?



70

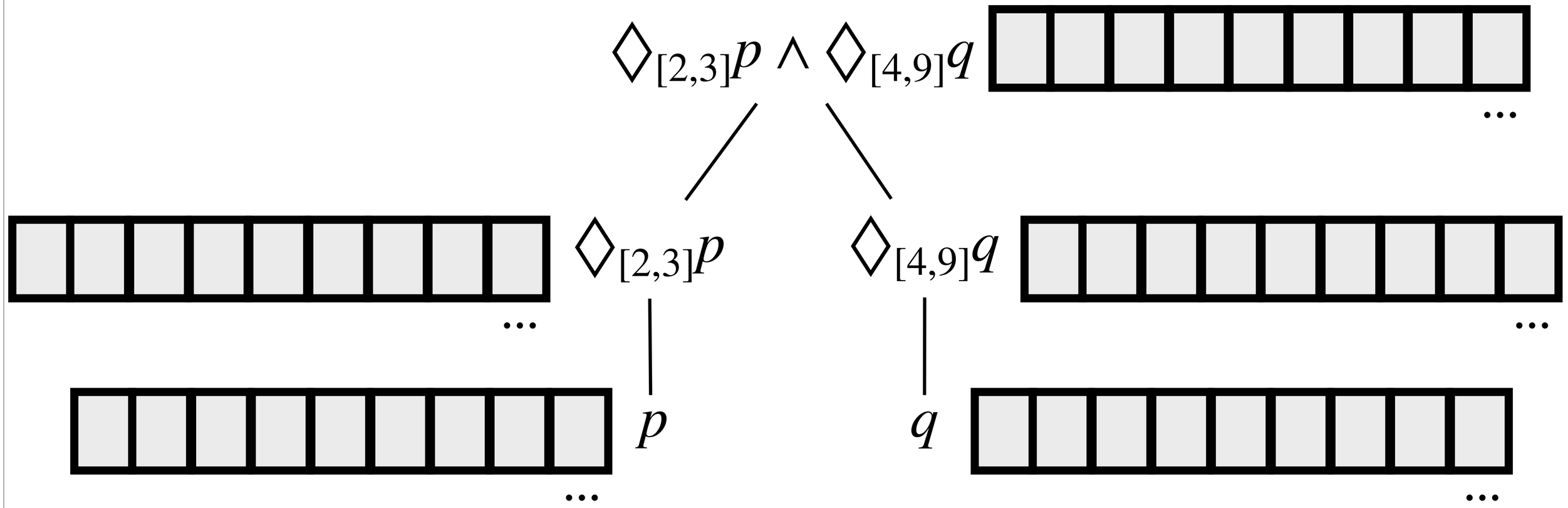
How much space does this require to monitor?



71

How much space does this require to monitor?

No siblings -> No buffer



72

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi$$

73

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi$$

$$\equiv$$

$$\Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2)$$
$$u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$

74

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi$$

$\equiv$ ***"Factor out" the smallest shared time window***

$$\Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2)$$
$$u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$

75

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi$$

$$\equiv$$

"Factor out" the smallest shared time window

$$\Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2)$$
$$u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$

76

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi$$

$\equiv$ ***"Factor out" the smallest shared time window***

$$\Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2)$$
$$u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$

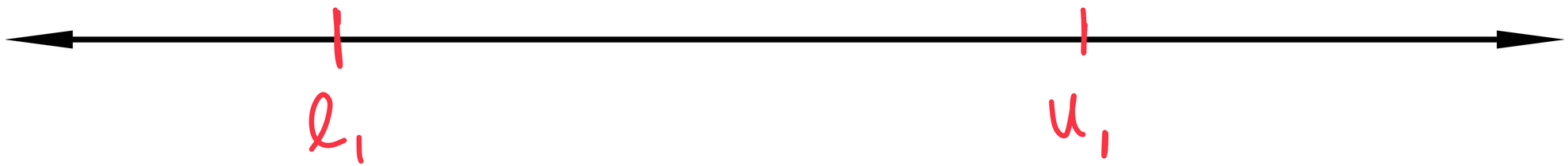
77

$$\square_{[l_1,u_1]}\varphi \wedge \square_{[l_2,u_2]}\psi$$

$\equiv$ ***"Factor out" the smallest shared time window***

$$\square_{[l_3,u_3]}(\square_{[l_1-l_3,u_1-u_3]}\varphi \wedge \square_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2)$$
$$u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$



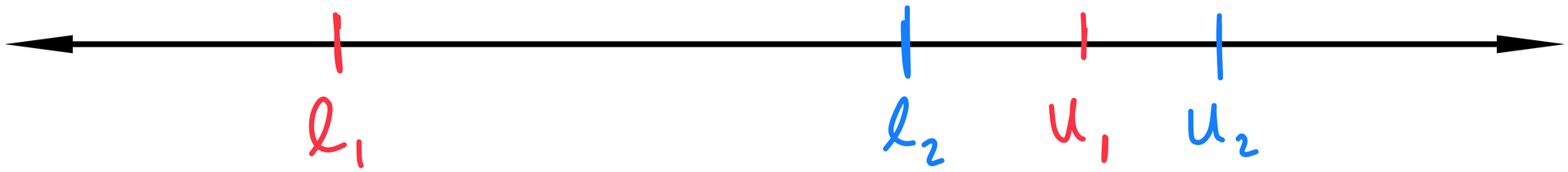
78

$$\square_{[l_1,u_1]}\varphi \wedge \square_{[l_2,u_2]}\psi$$

$\equiv$ ***"Factor out" the smallest shared time window***

$$\square_{[l_3,u_3]}(\square_{[l_1-l_3,u_1-u_3]}\varphi \wedge \square_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2)$$
$$u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$



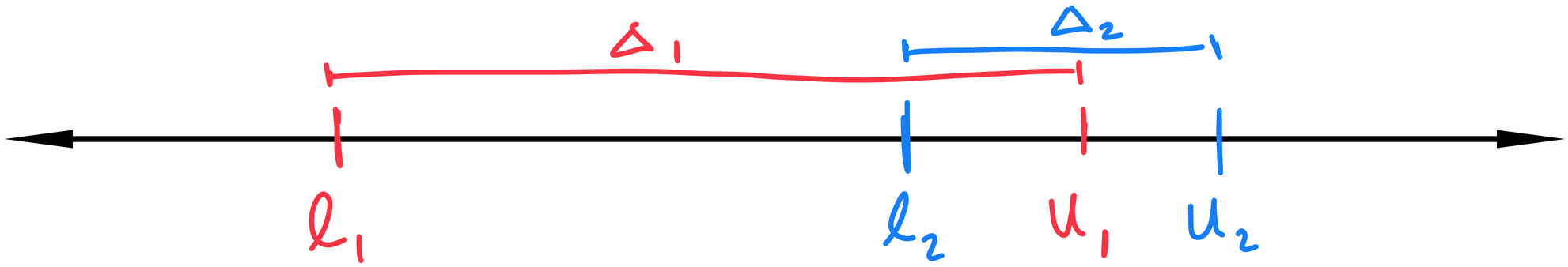
79

$$\square_{[l_1,u_1]}\varphi \wedge \square_{[l_2,u_2]}\psi$$

$\equiv$ ***"Factor out" the smallest shared time window***

$$\square_{[l_3,u_3]}(\square_{[l_1-l_3,u_1-u_3]}\varphi \wedge \square_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2)$$
$$u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$



80

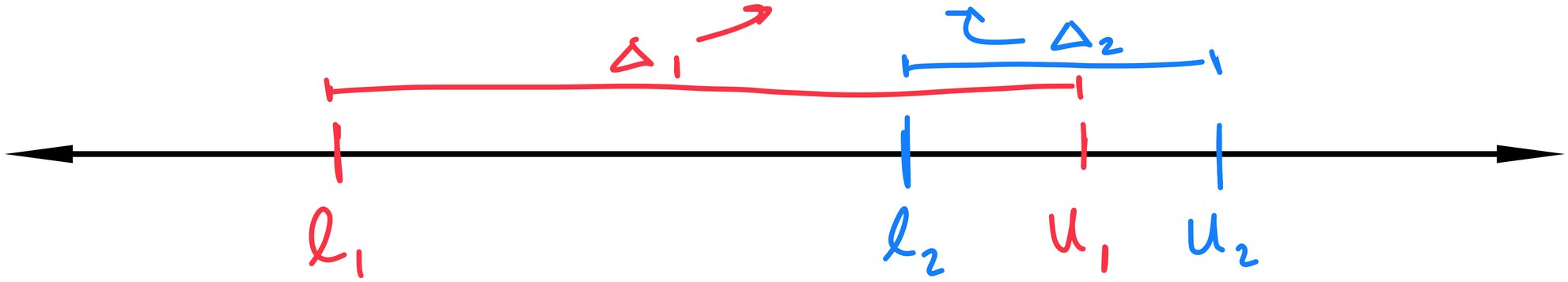
$$\square_{[l_1,u_1]}\varphi \wedge \square_{[l_2,u_2]}\psi$$

$\equiv$

"Factor out" the smallest shared time window

$$\square_{[l_3,u_3]}(\square_{[l_1-l_3,u_1-u_3]}\varphi \wedge \square_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2)$$
$$u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$



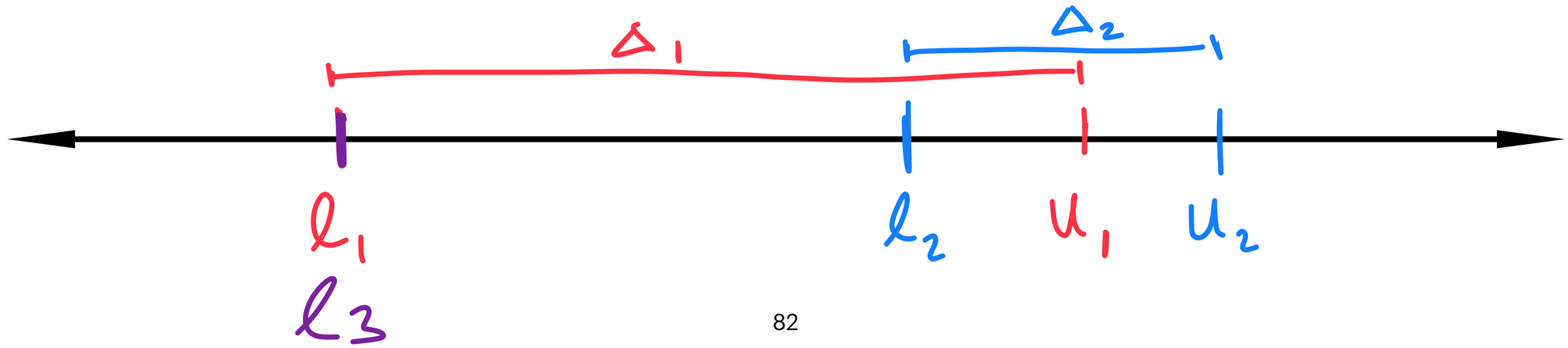
81

$$\square_{[l_1,u_1]}\varphi \wedge \square_{[l_2,u_2]}\psi$$

$\equiv$ **"Factor out" the smallest shared time window**

$$\square_{[l_3,u_3]}(\square_{[l_1-l_3,u_1-u_3]}\varphi \wedge \square_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2)$$
$$u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$



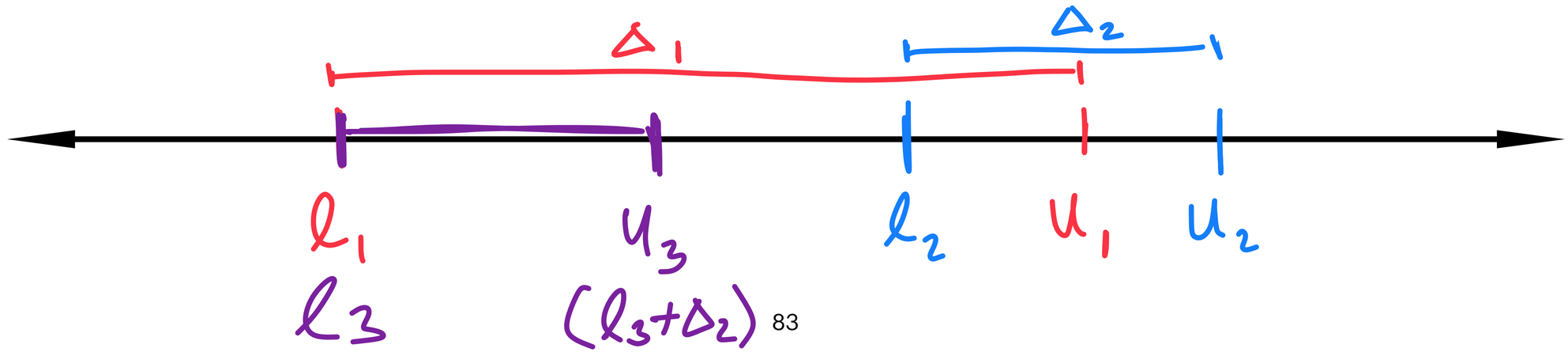
82

$$\square_{[l_1,u_1]}\varphi \wedge \square_{[l_2,u_2]}\psi$$

$\equiv$ **"Factor out" the smallest shared time window**

$$\square_{[l_3,u_3]}(\square_{[l_1-l_3,u_1-u_3]}\varphi \wedge \square_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2)$$
$$u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$



83

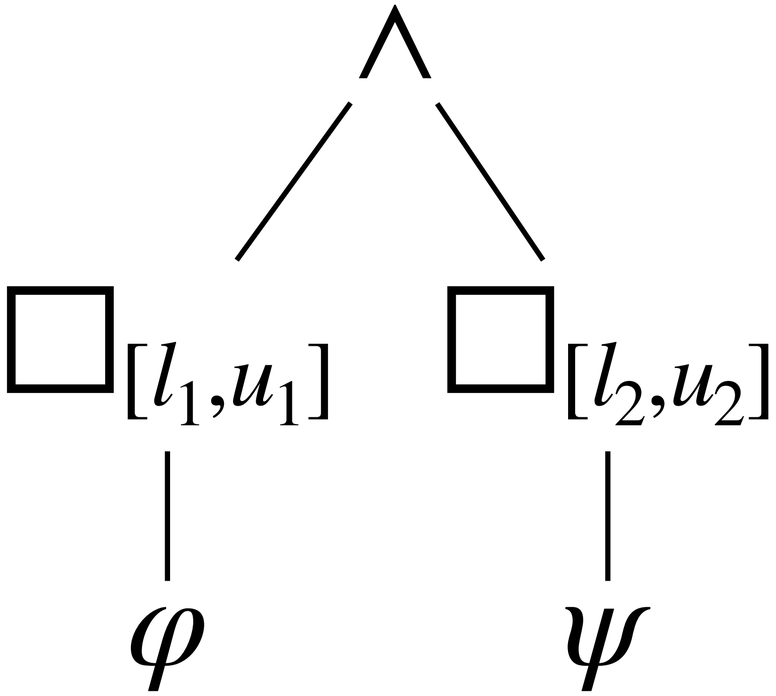
$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \quad \equiv \quad \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2) \qquad u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$

84

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \quad \equiv \quad \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

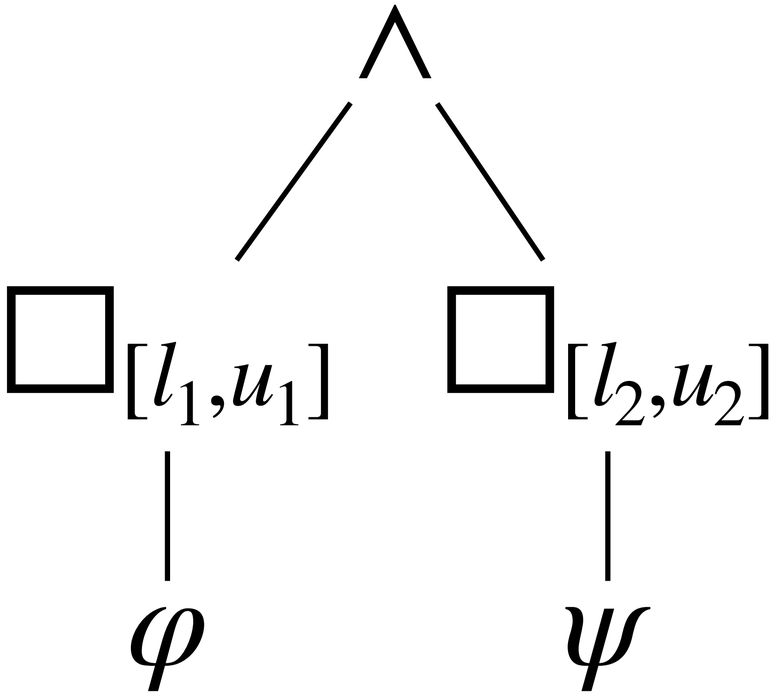
$$l_3 = min(l_1, l_2) \qquad u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$

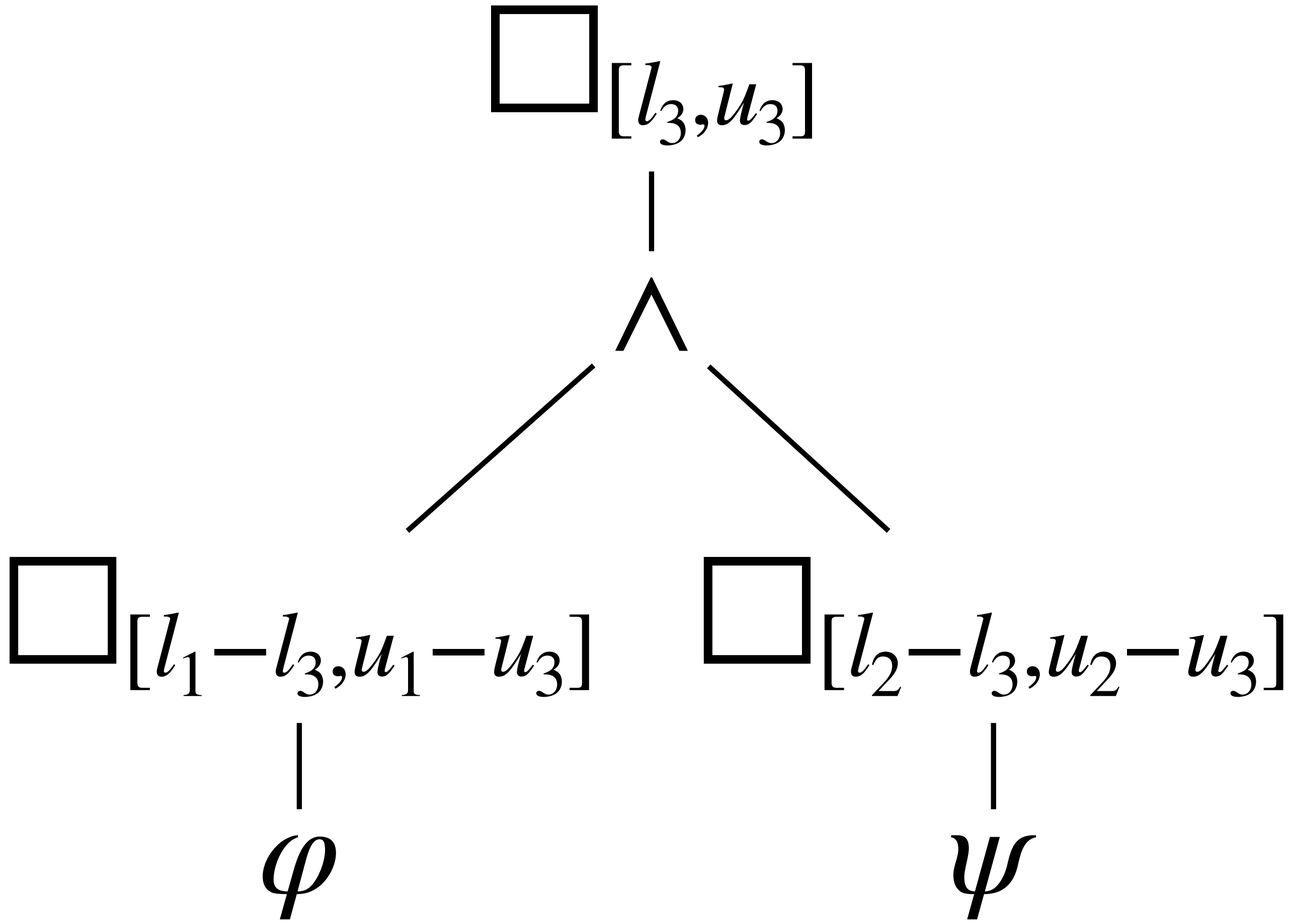


85

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \quad \equiv \quad \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2) \qquad u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$

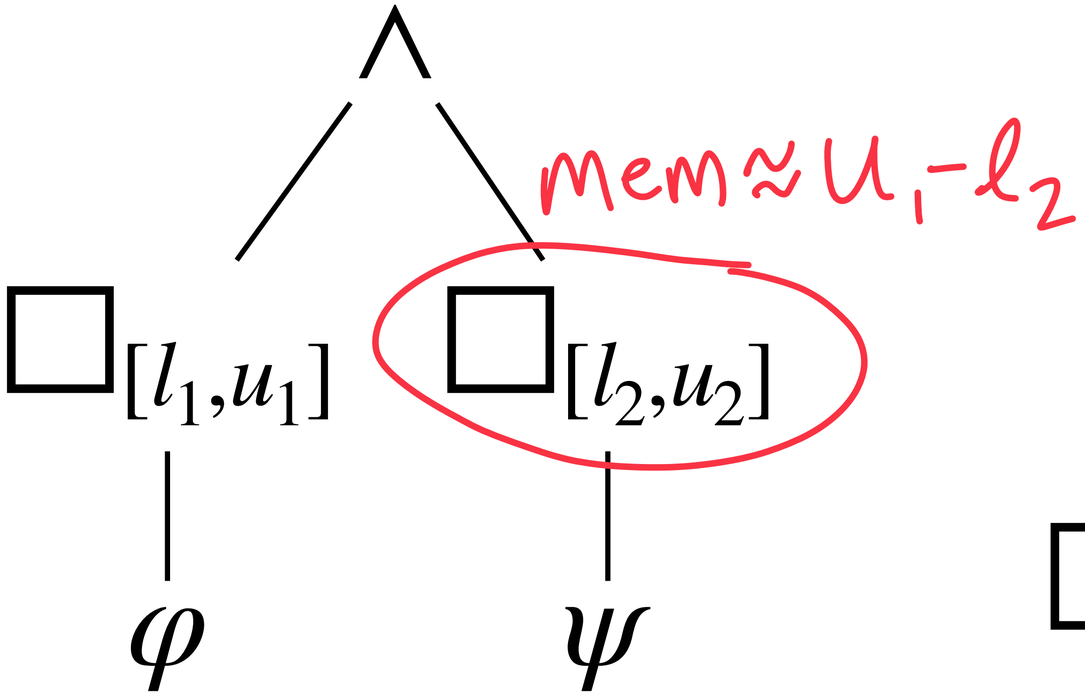


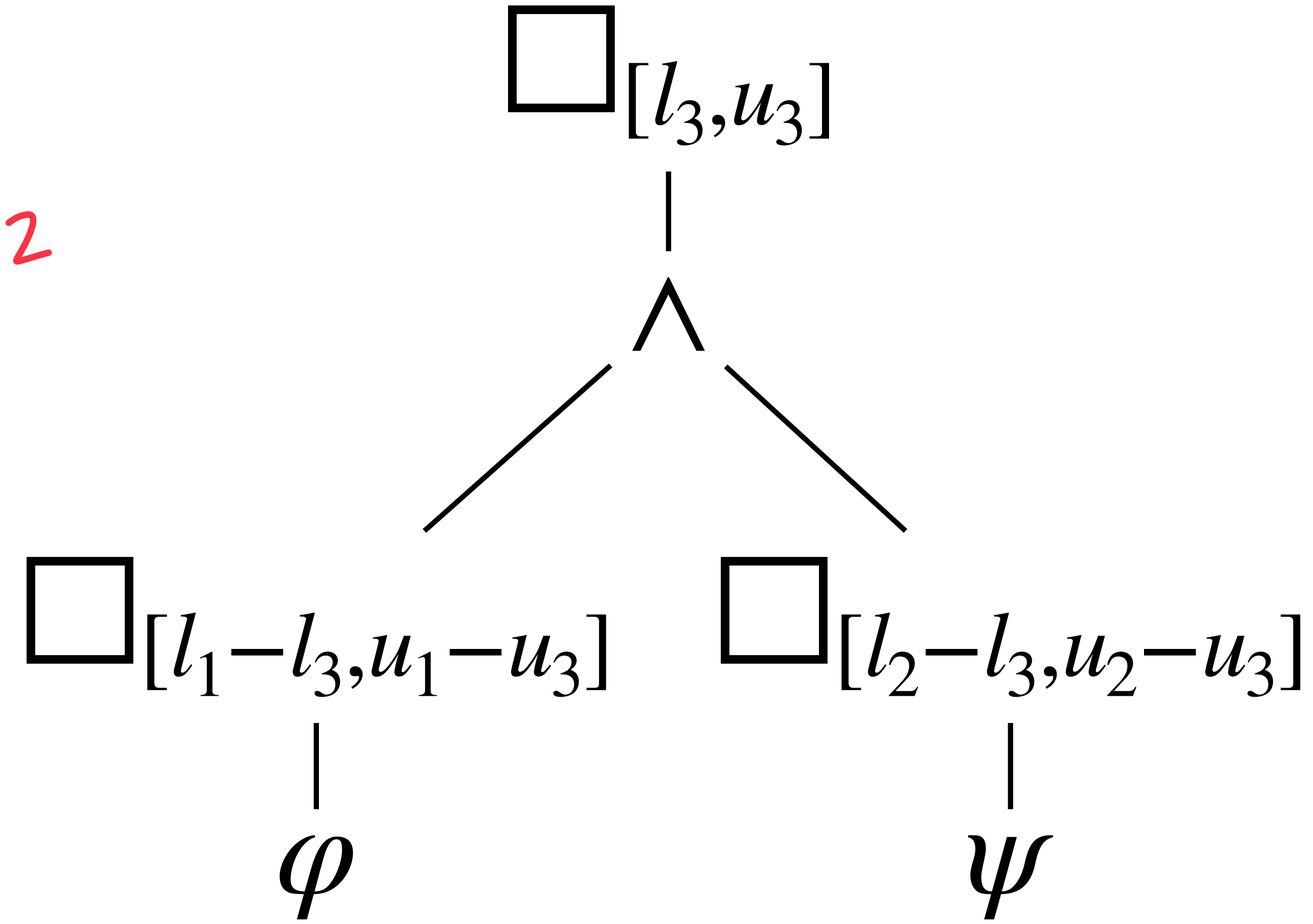


86

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \quad \equiv \quad \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2) \qquad u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$

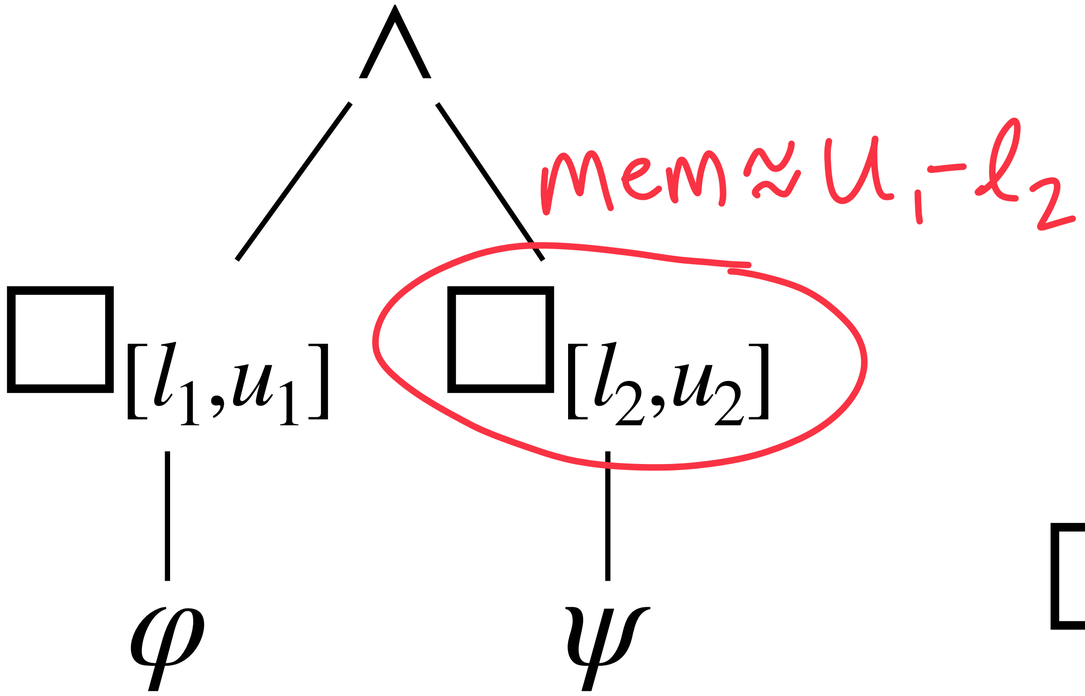


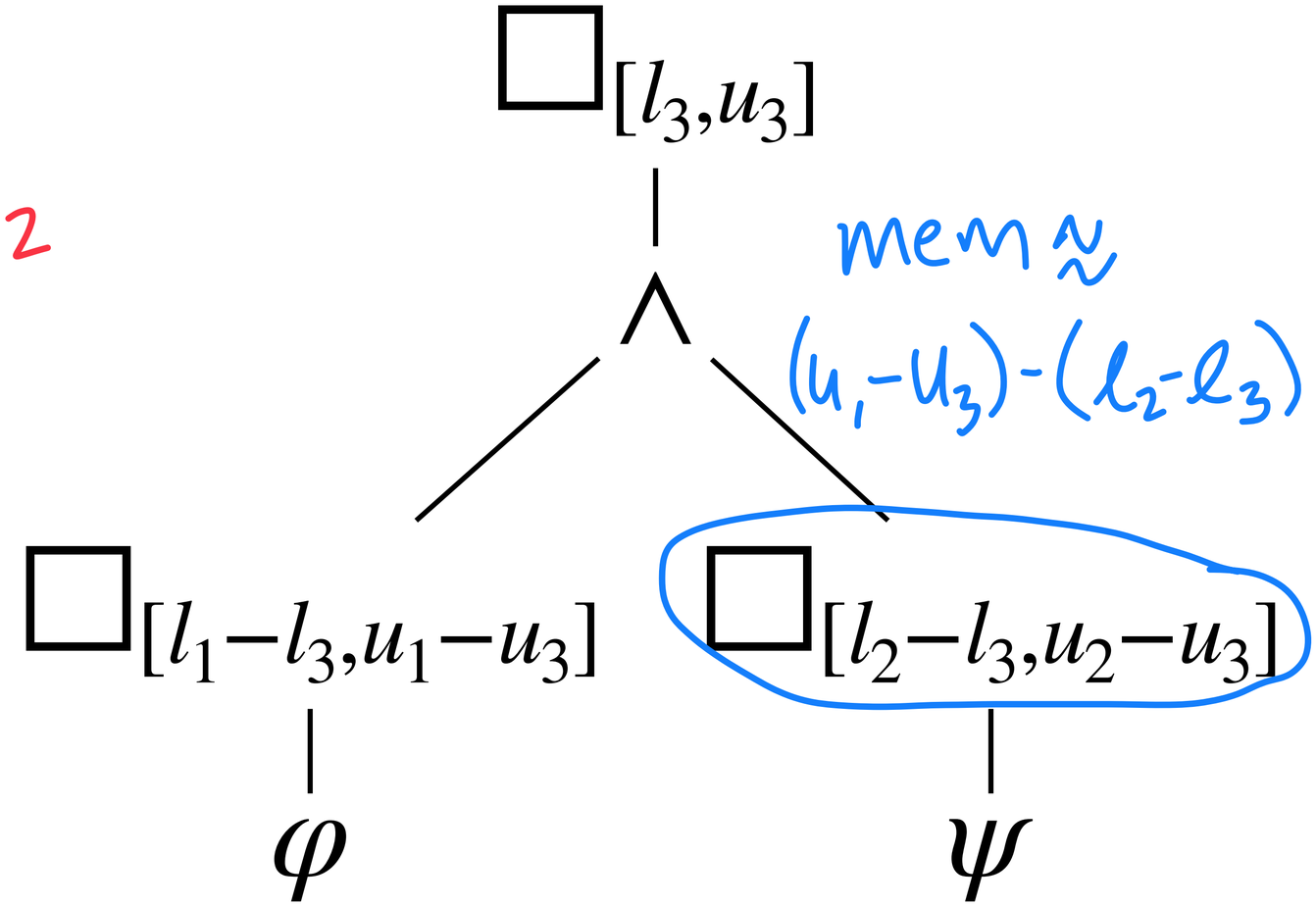


87

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \quad \equiv \quad \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2) \qquad u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$



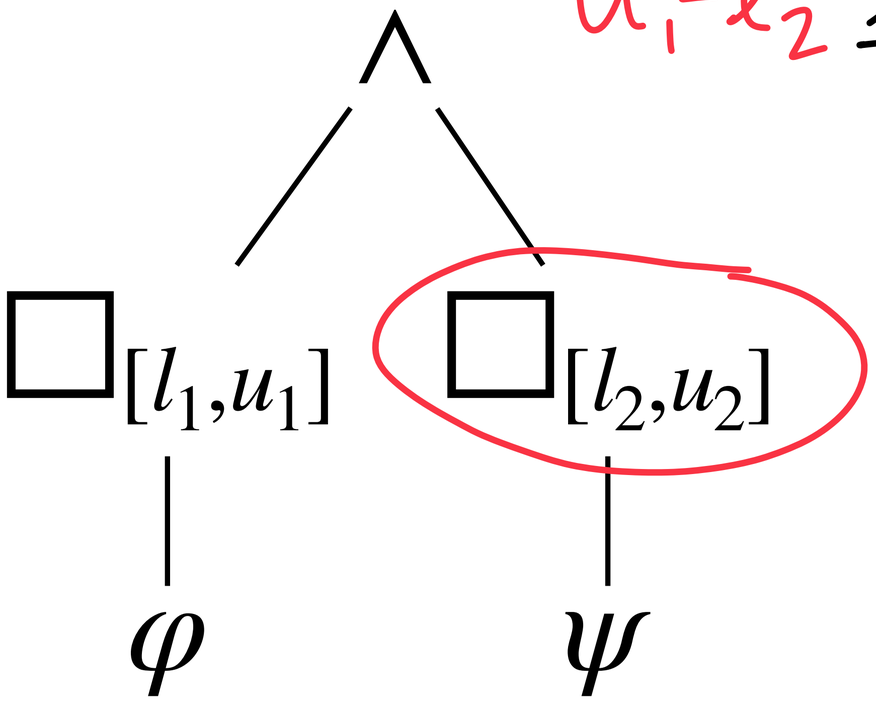


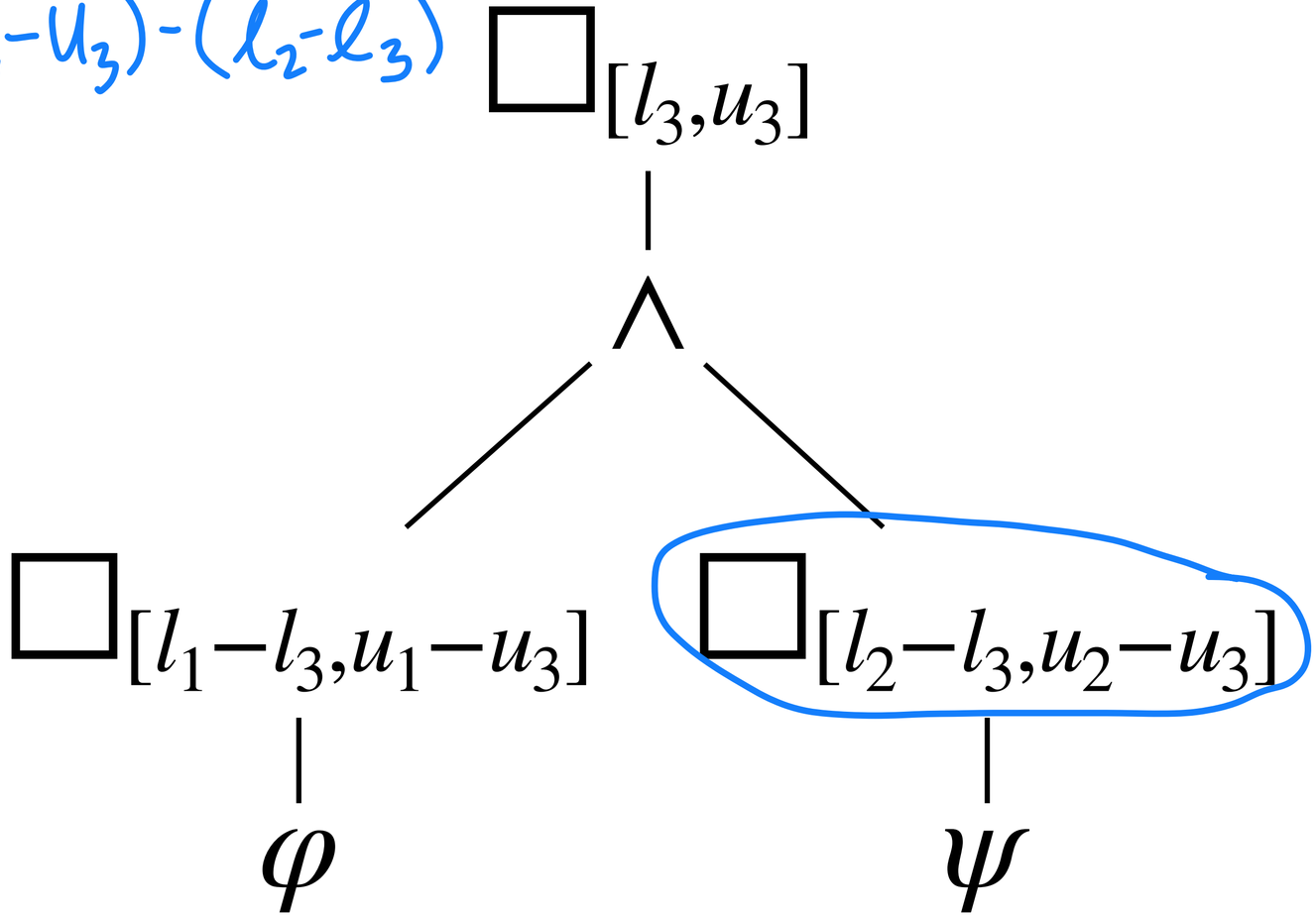
88

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \quad \equiv \quad \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2) \qquad u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$

$u_1 - l_2 \geq (u_1 - u_3) - (l_2 - l_3)$



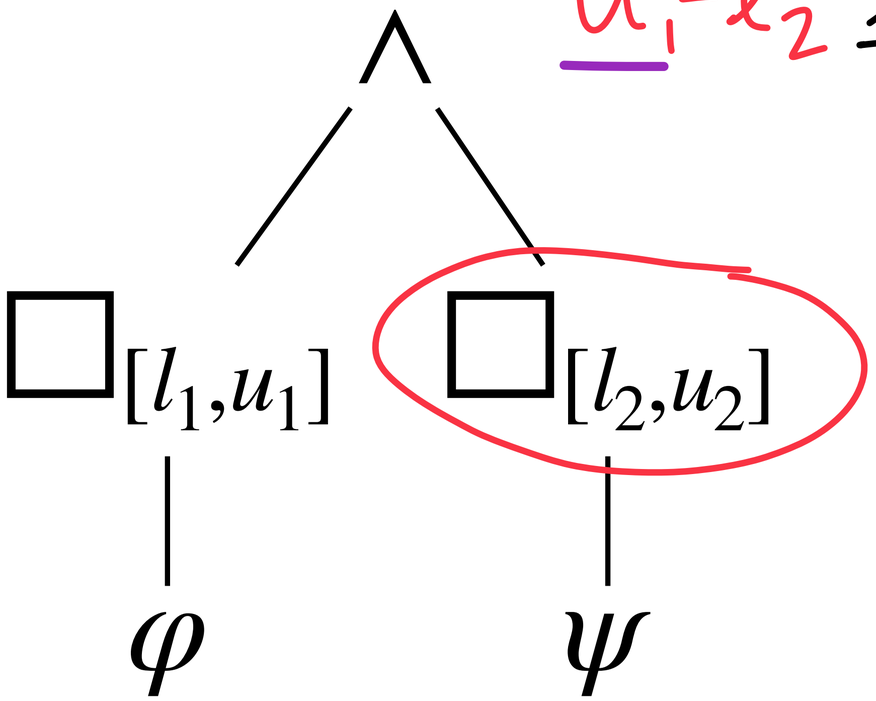


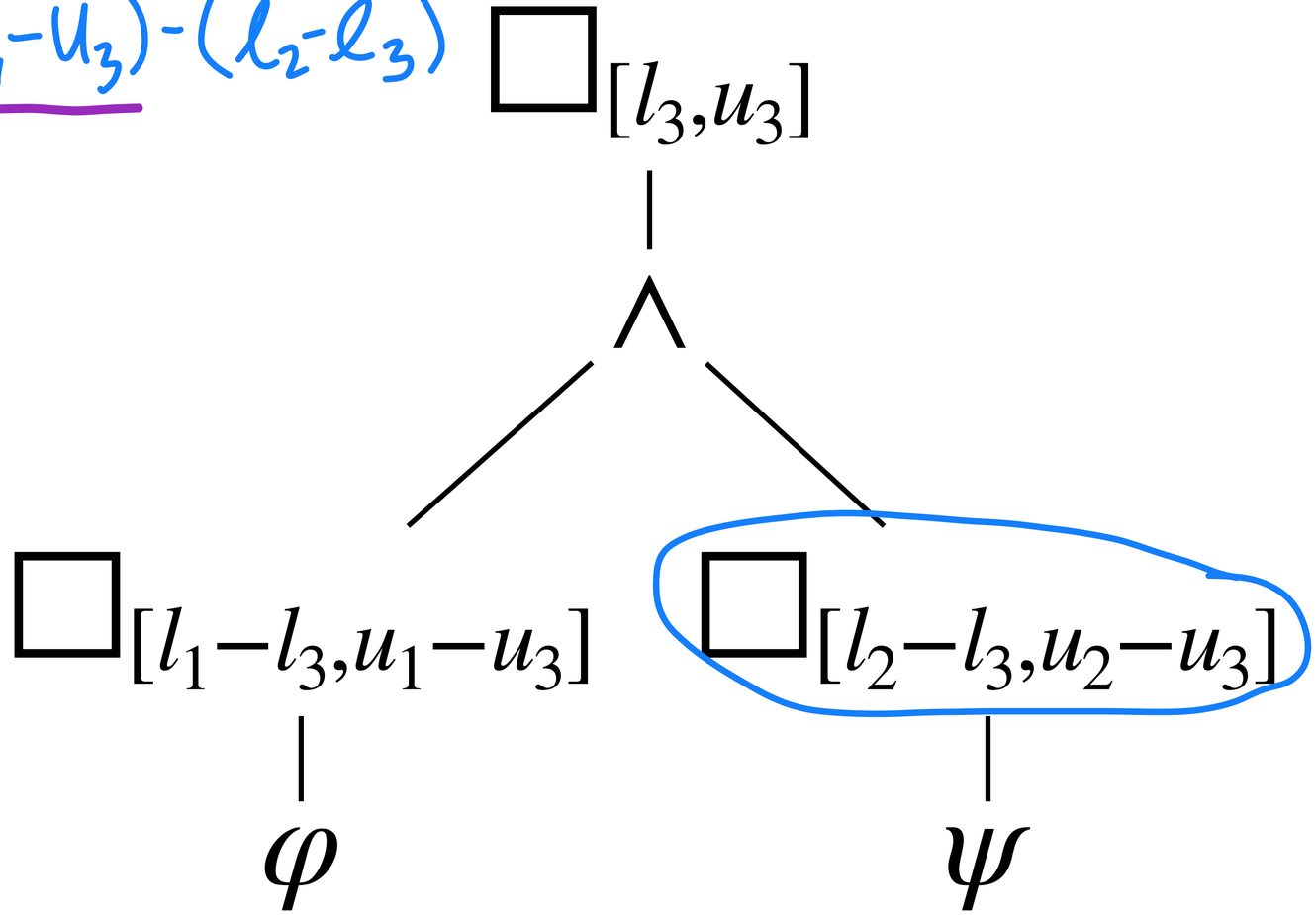
89

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \quad \equiv \quad \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2) \qquad u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$

$u_1 - l_2 \geq (u_1 - u_3) - (l_2 - l_3)$



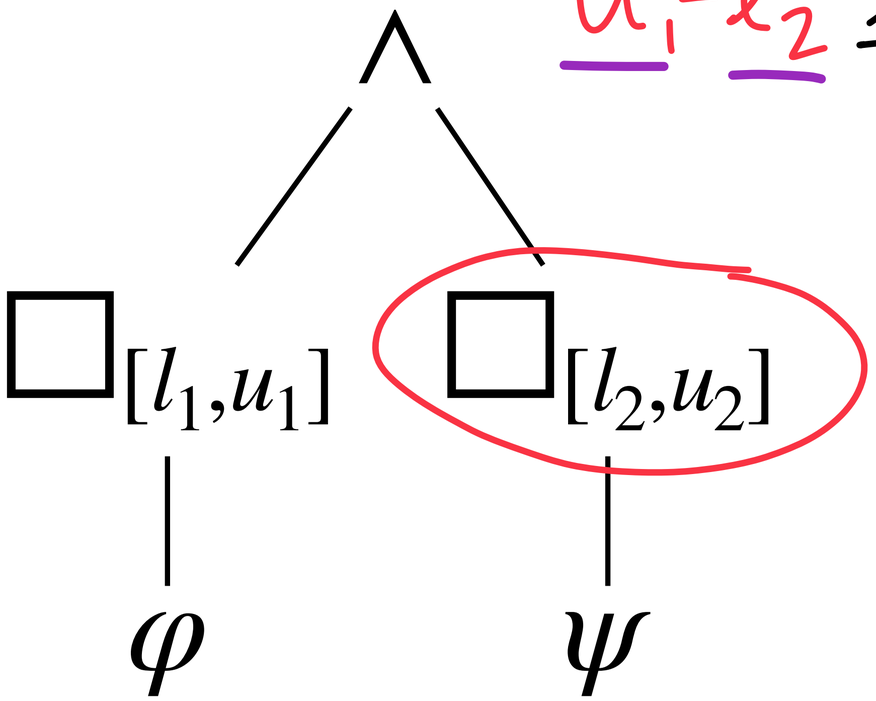


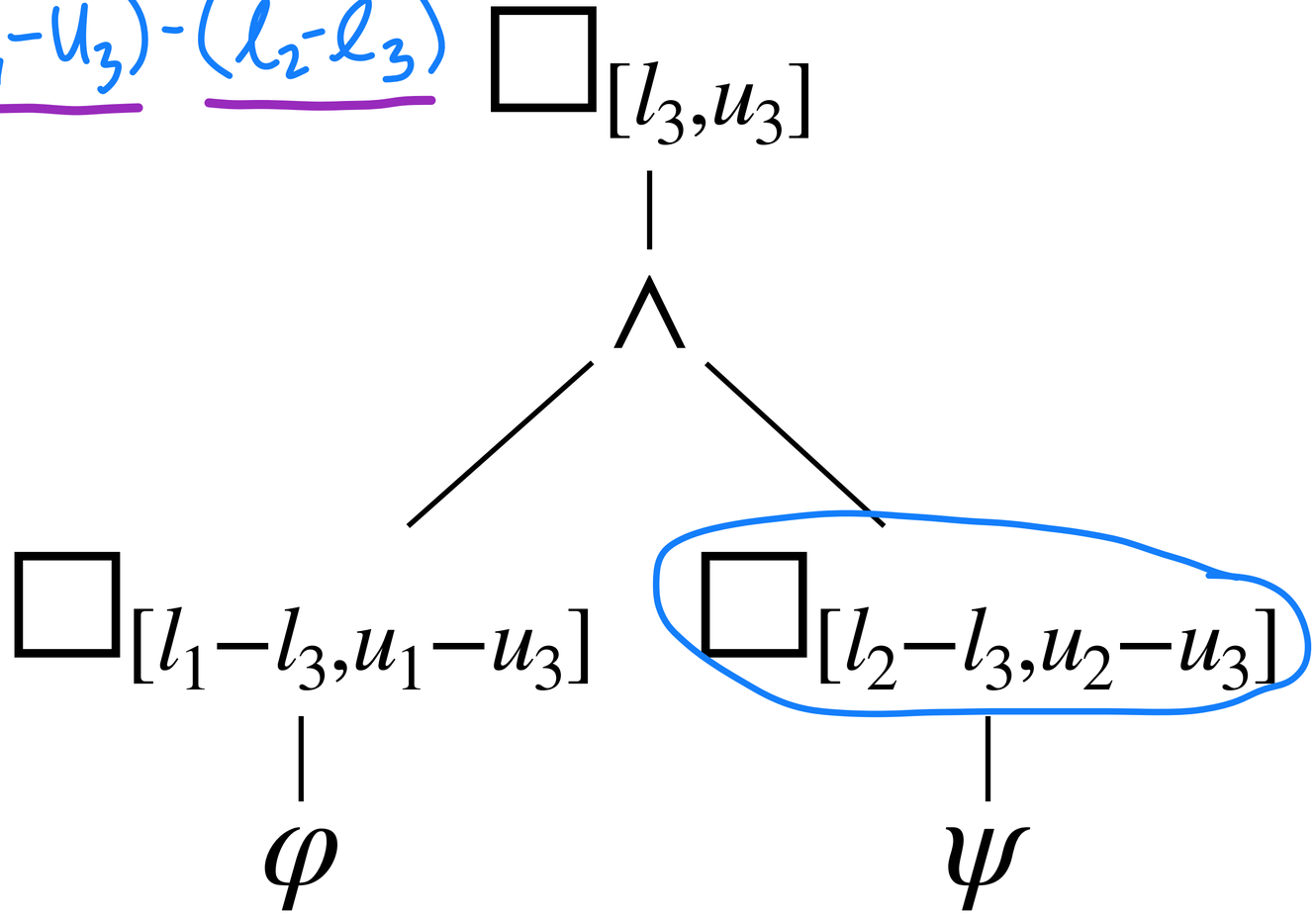
90

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \quad \equiv \quad \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2) \qquad u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$

$u_1 - l_2 \geq (u_1 - u_3) - (l_2 - l_3)$



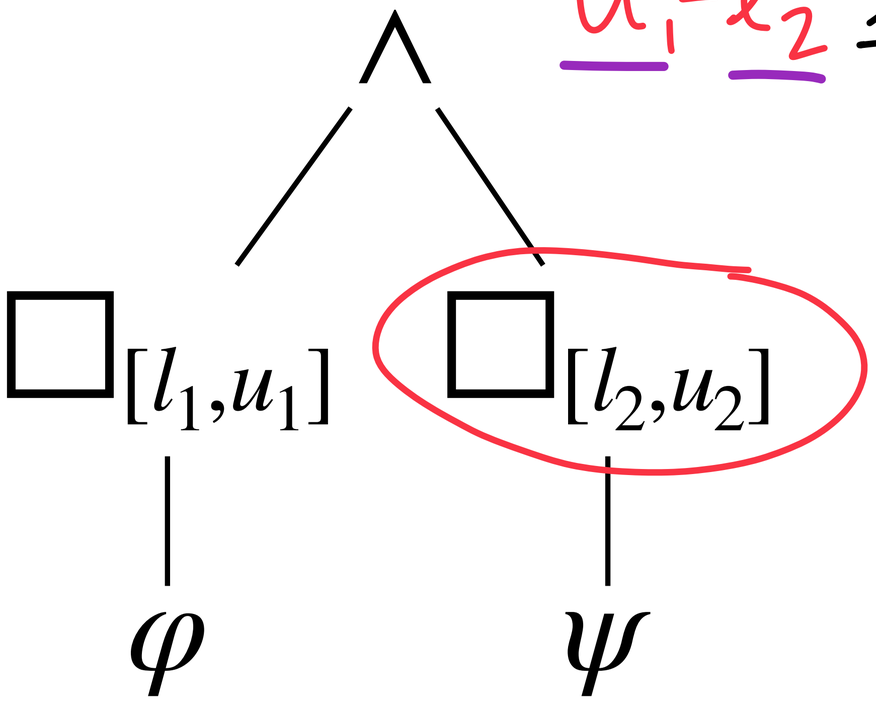


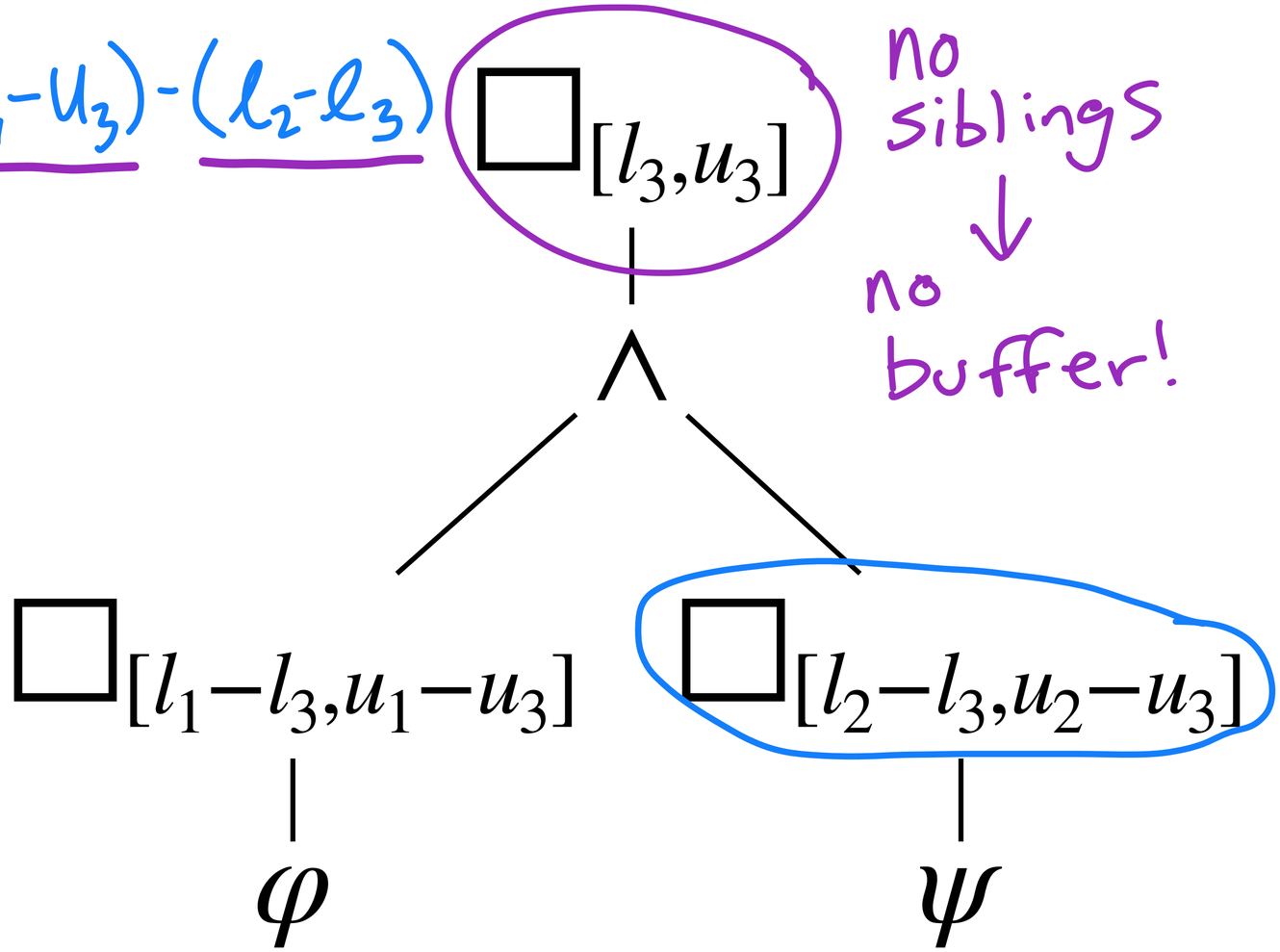
91

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \quad \equiv \quad \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

$$l_3 = min(l_1, l_2) \qquad u_3 = l_3 + min(u_1 - l_1, u_2 - l_2)$$

$u_1 - l_2 \geq (u_1 - u_3) - (l_2 - l_3)$





no siblings → no buffer!

92

Most rules were inspired by LTL equivalences....

93

Most rules were inspired by LTL equivalences....

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi$$
$$\equiv$$
$$\Box_{[l_3,u_3]}\left(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi\right)$$

inspired by $\Box\varphi \wedge \Box\psi \equiv \Box(\varphi \wedge \psi)$

94

Most rules were inspired by LTL equivalences....

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \equiv \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

inspired by $\Box\varphi \wedge \Box\psi \equiv \Box(\varphi \wedge \psi)$

$$\Box_{[l_1,u_1]}\Box_{[l_2,u_2]}\varphi \equiv \Box_{[l_1+l_2,u_1+u_2]}\varphi$$

inspired by $\Box\Box\varphi \equiv \Box\varphi$

95

Most rules were inspired by LTL equivalences....

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \equiv \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

inspired by $\Box\varphi \wedge \Box\psi \equiv \Box(\varphi \wedge \psi)$

$$\Box_{[l_1,u_1]}\Box_{[l_2,u_2]}\varphi \equiv \Box_{[l_1+l_2,u_1+u_2]}\varphi$$

inspired by $\Box\Box\varphi \equiv \Box\varphi$

$$(\varphi_1 \mathcal{U}_{[l,u_1]}\varphi_2) \wedge (\varphi_3 \mathcal{U}_{[l,u_2]}\varphi_2) \equiv (\varphi_1 \wedge \varphi_3)\mathcal{U}_{[l,u_1]}\varphi_2$$

inspired by $(\varphi_1 \mathcal{U}\varphi_2) \wedge (\varphi_3 \mathcal{U}\varphi_2) \equiv (\varphi_1 \wedge \varphi_3)\mathcal{U}\varphi_2$

96

Most rules were inspired by LTL equivalences....

$$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \equiv \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$$

inspired by $\Box\varphi \wedge \Box\psi \equiv \Box(\varphi \wedge \psi)$

$$\Box_{[l_1,u_1]}\Box_{[l_2,u_2]}\varphi \equiv \Box_{[l_1+l_2,u_1+u_2]}\varphi$$

inspired by $\Box\Box\varphi \equiv \Box\varphi$

$$(\varphi_1 \mathcal{U}_{[l,u_1]}\varphi_2) \wedge (\varphi_3 \mathcal{U}_{[l,u_2]}\varphi_2) \equiv (\varphi_1 \wedge \varphi_3)\mathcal{U}_{[l,u_1]}\varphi_2$$

inspired by $(\varphi_1 \mathcal{U}\varphi_2) \wedge (\varphi_3 \mathcal{U}\varphi_2) \equiv (\varphi_1 \wedge \varphi_3)\mathcal{U}\varphi_2$

Does every LTL equiv. have a corollary in MLTL?

Does every LTL equiv. have a corollary in MLTL?

$$\Diamond(\varphi \mathcal{U} \psi) \equiv \Diamond \psi$$

98

Does every LTL equiv. have a corollary in MLTL?

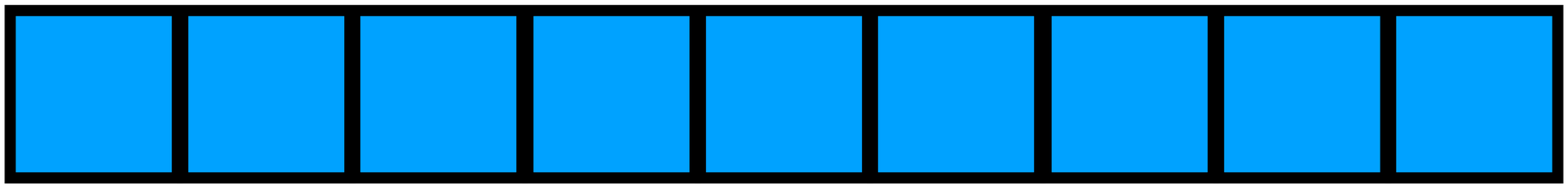
$$\Diamond(\varphi \mathcal{U} \psi) \equiv \Diamond \psi$$

$$\Diamond_{[a,b]}(\varphi \mathcal{U}_{[c,d]} \psi) \overset{?}{\equiv} \Diamond_{[?,?]} \psi$$

99

Does every LTL equiv. have a corollary in MLTL?

$$\Diamond(\varphi \mathcal{U} \psi) \equiv \Diamond \psi$$



100

Does every LTL equiv. have a corollary in MLTL?

$$\Diamond(\varphi \mathcal{U} \psi) \equiv \Diamond \psi$$

101

Does every LTL equiv. have a corollary in MLTL?

$$\overset{?}{\models} \quad \Diamond(\varphi \mathcal{U} \psi) \equiv \Diamond \psi \quad \overset{?}{\nexists}$$

102

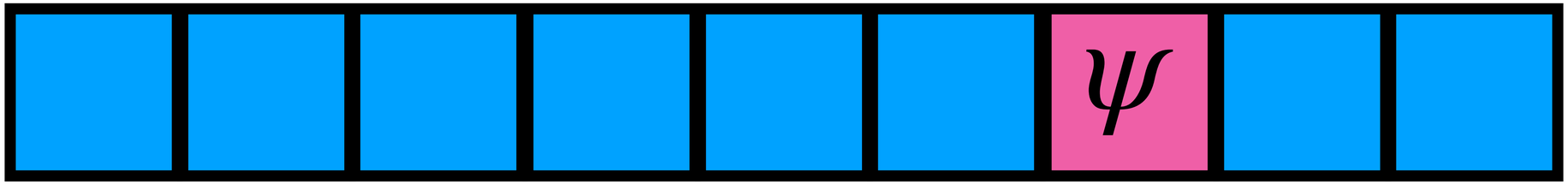
Does every LTL equiv. have a corollary in MLTL?

$$\models \quad \Diamond(\varphi \mathcal{U} \psi) \equiv \Diamond \psi \quad \nexists$$

103

Does every LTL equiv. have a corollary in MLTL?

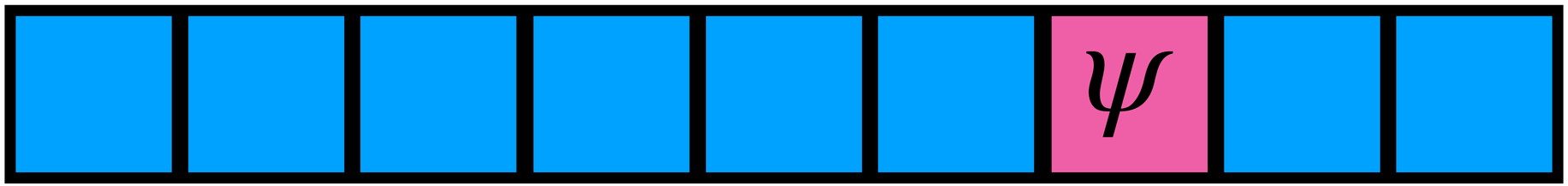
$$\Diamond(\varphi \mathcal{U} \psi) \equiv \Diamond \psi$$



104

Does every LTL equiv. have a corollary in MLTL?

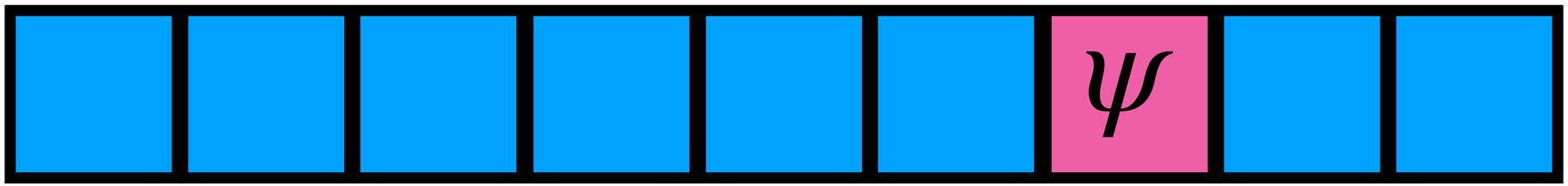
$\overset{?}{\models} \quad \Diamond(\varphi \mathcal{U} \psi) \equiv \Diamond \psi \quad \overset{?}{\not\models}$



105

Does every LTL equiv. have a corollary in MLTL?

$$\models \quad \Diamond(\varphi \mathcal{U} \psi) \equiv \Diamond \psi \quad \exists\!\models$$



106

Does every LTL equiv. have a corollary in MLTL?

$$\Diamond(\varphi \mathcal{U} \psi) \equiv \Diamond \psi$$

$$\Diamond_{[a,b]}(\varphi \mathcal{U}_{[c,d]} \psi) \overset{?}{\equiv} \Diamond_{[?,?]} \psi$$

107

Does every LTL equiv. have a corollary in MLTL?

$$\Diamond_{[a,b]}(\varphi \mathcal{U}_{[c,d]} \psi) \overset{?}{\equiv} \Diamond_{[?,?]} \psi$$

Does every LTL equiv. have a corollary in MLTL?

$$\Diamond_{[0,2]}(p \mathcal{U}_{[0,3]} q) \overset{?}{\equiv} \Diamond_{[?,?]} q$$

109

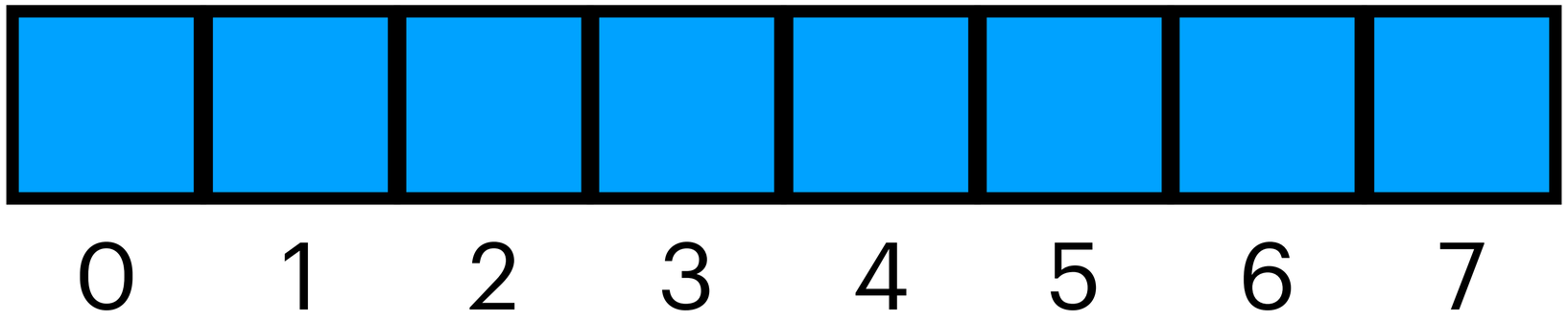
Does every LTL equiv. have a corollary in MLTL?

$$\Diamond_{[0,2]}(p \mathcal{U}_{[0,3]} q) \overset{?}{\equiv} \Diamond_{[0,3]} q$$

110

Does every LTL equiv. have a corollary in MLTL?

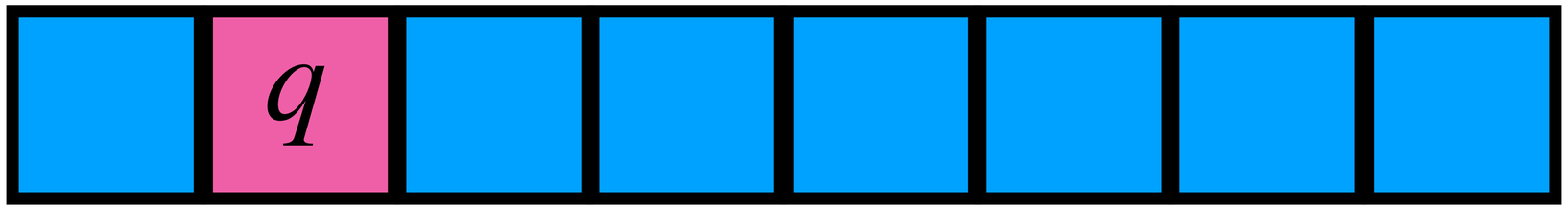
$$\Diamond_{[0,2]}(p \mathcal{U}_{[0,3]} q) \equiv \Diamond_{[0,3]} q$$



111

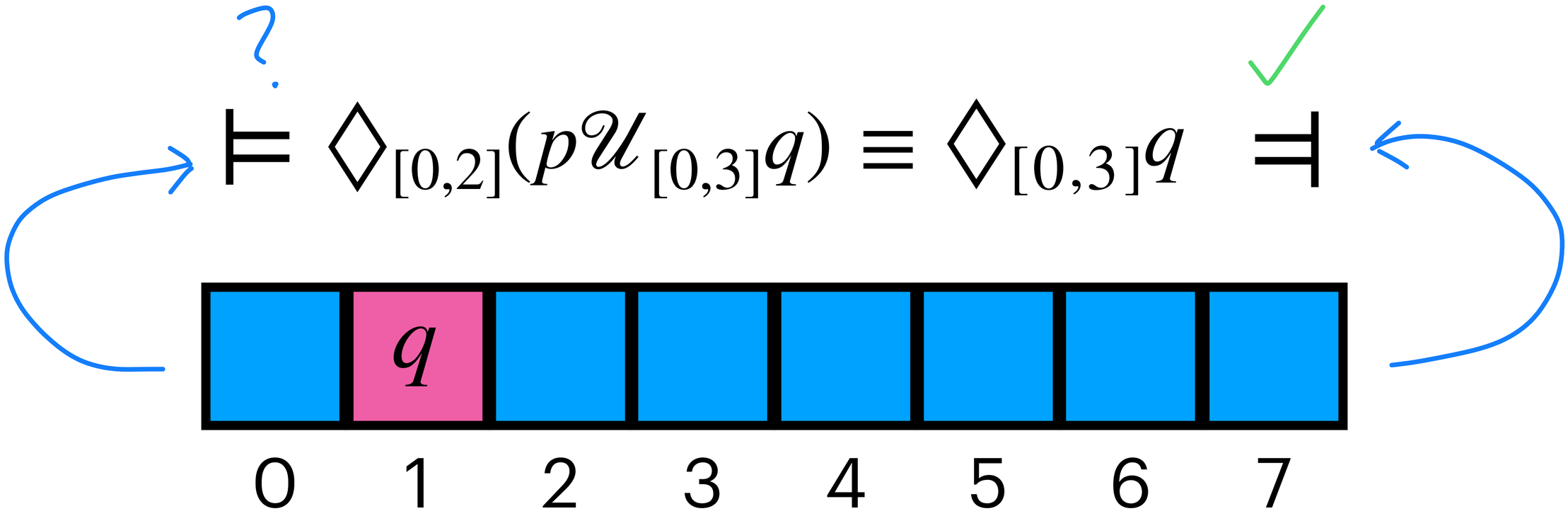
Does every LTL equiv. have a corollary in MLTL?

$$\models \Diamond_{[0,2]}(p \mathcal{U}_{[0,3]} q) \equiv \Diamond_{[0,3]} q \quad \not\models$$



112

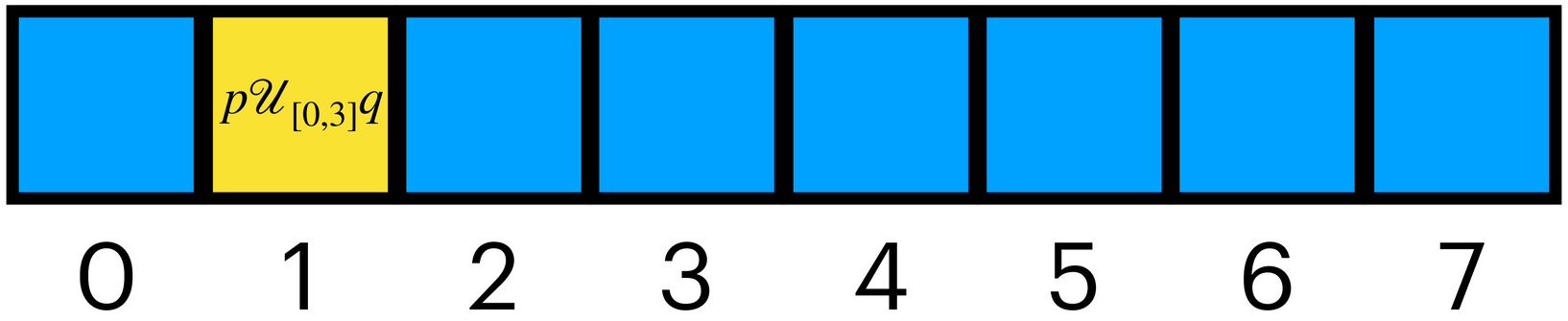
Does every LTL equiv. have a corollary in MLTL?



113

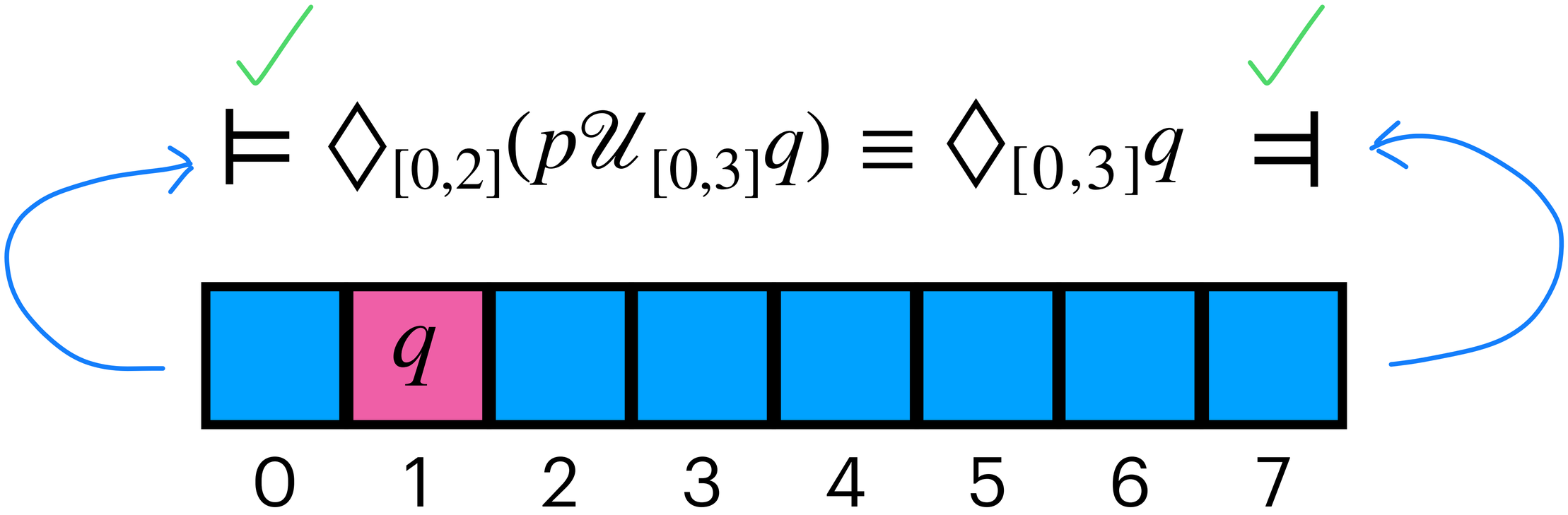
Does every LTL equiv. have a corollary in MLTL?

$$\models \Diamond_{[0,2]}(p \mathcal{U}_{[0,3]} q) \equiv \Diamond_{[0,3]} q \quad \not\models$$



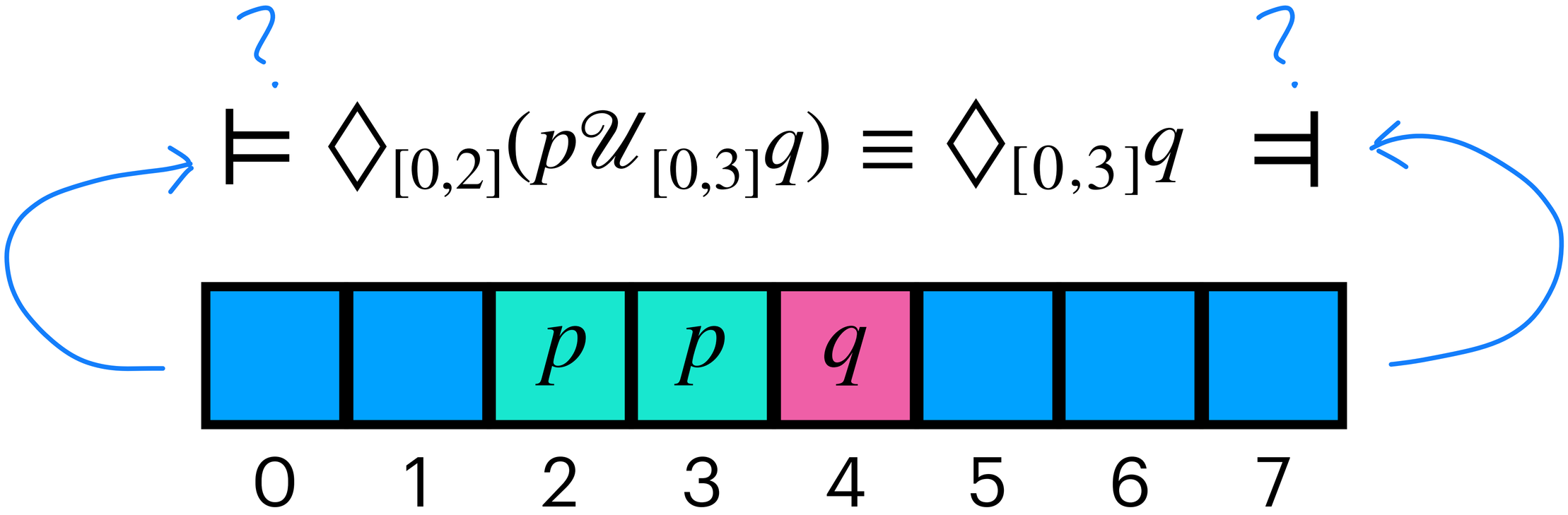
114

Does every LTL equiv. have a corollary in MLTL?



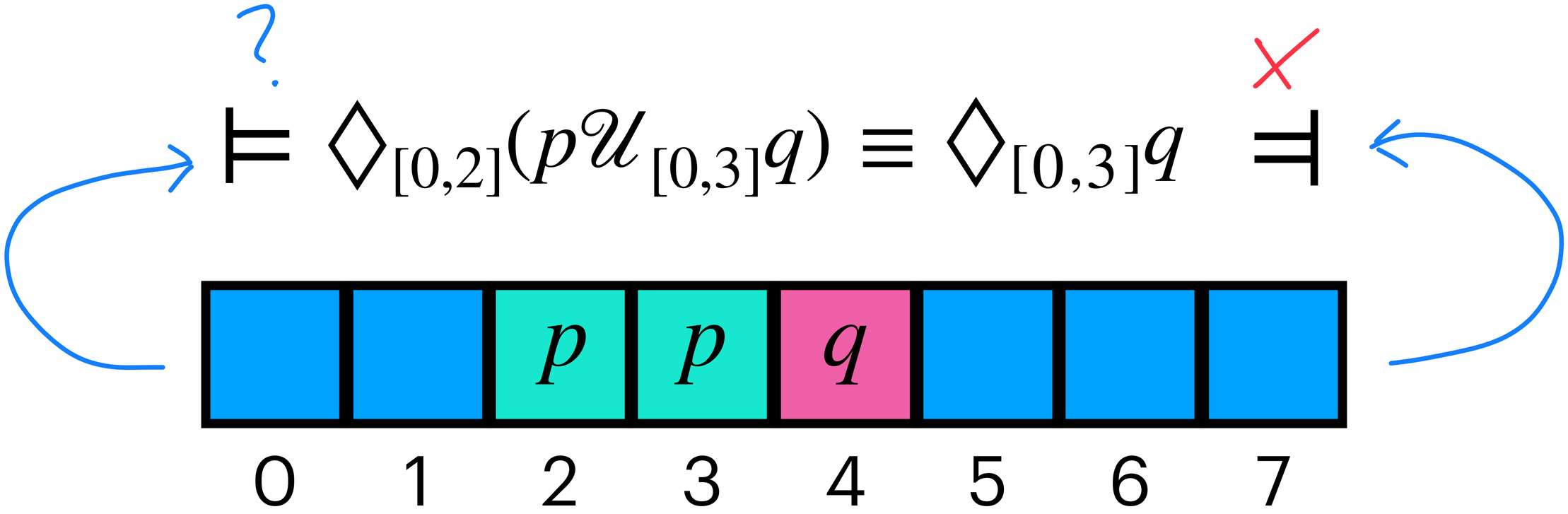
115

Does every LTL equiv. have a corollary in MLTL?



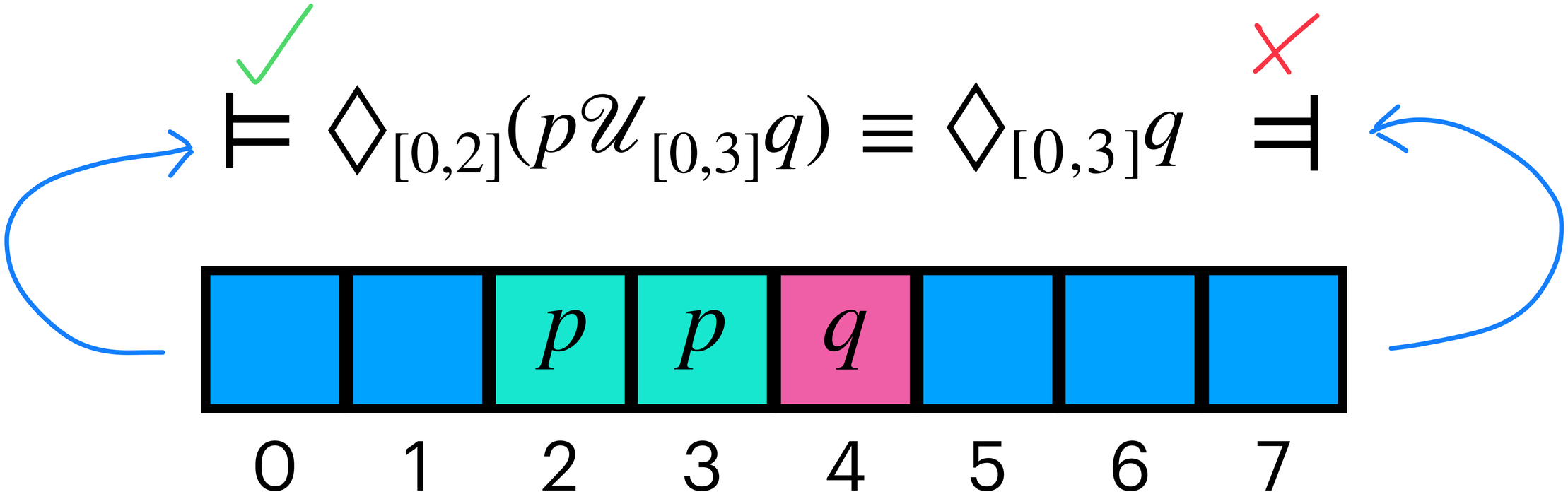
116

Does every LTL equiv. have a corollary in MLTL?



117

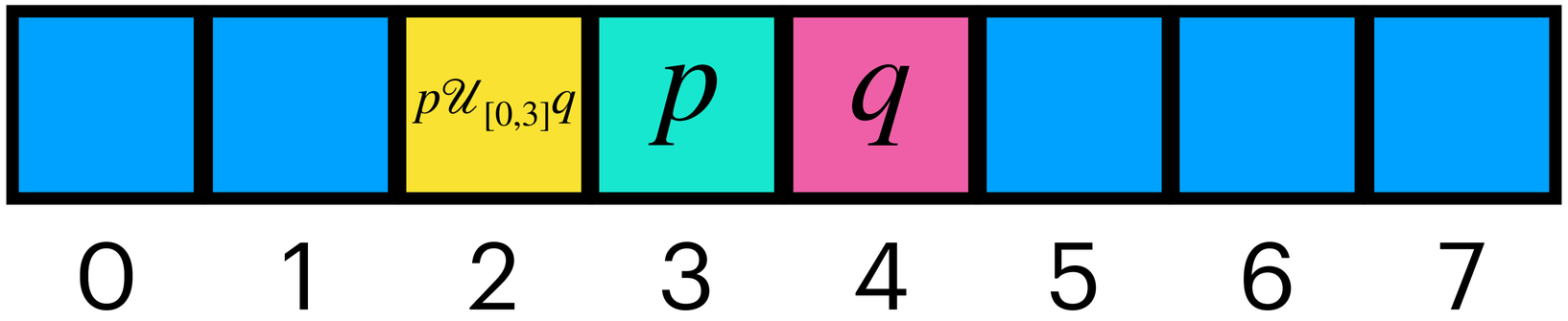
Does every LTL equiv. have a corollary in MLTL?



118

Does every LTL equiv. have a corollary in MLTL?

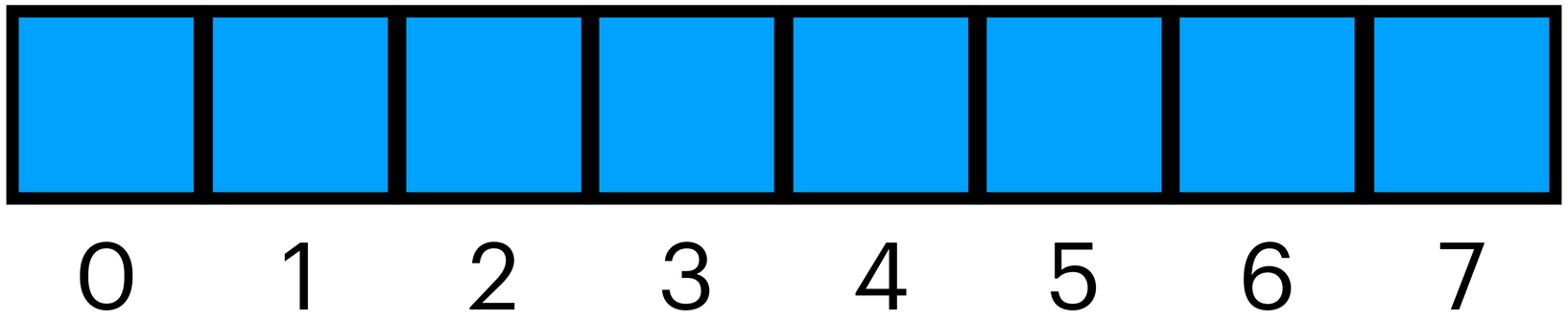
$$\models \Diamond_{[0,2]}(p \mathcal{U}_{[0,3]} q) \equiv \Diamond_{[0,3]} q \quad \not\equiv$$



119

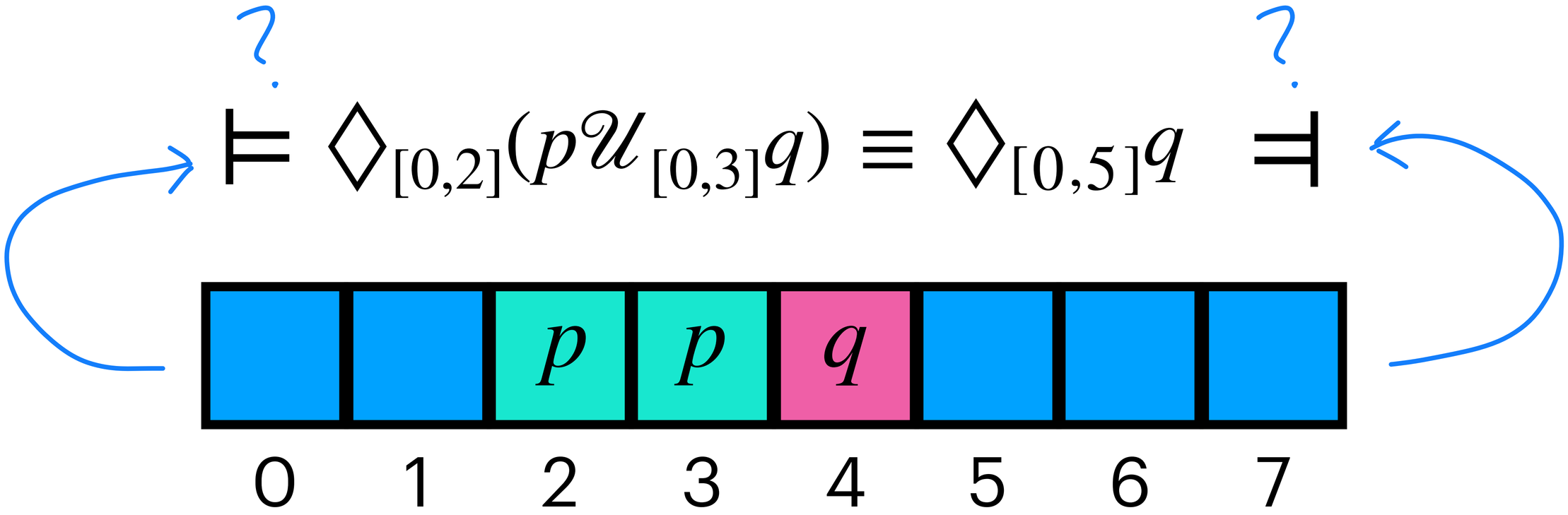
Does every LTL equiv. have a corollary in MLTL?

$$\Diamond_{[0,2]}(p \mathcal{U}_{[0,3]} q) \equiv \Diamond_{[0,5]} q$$



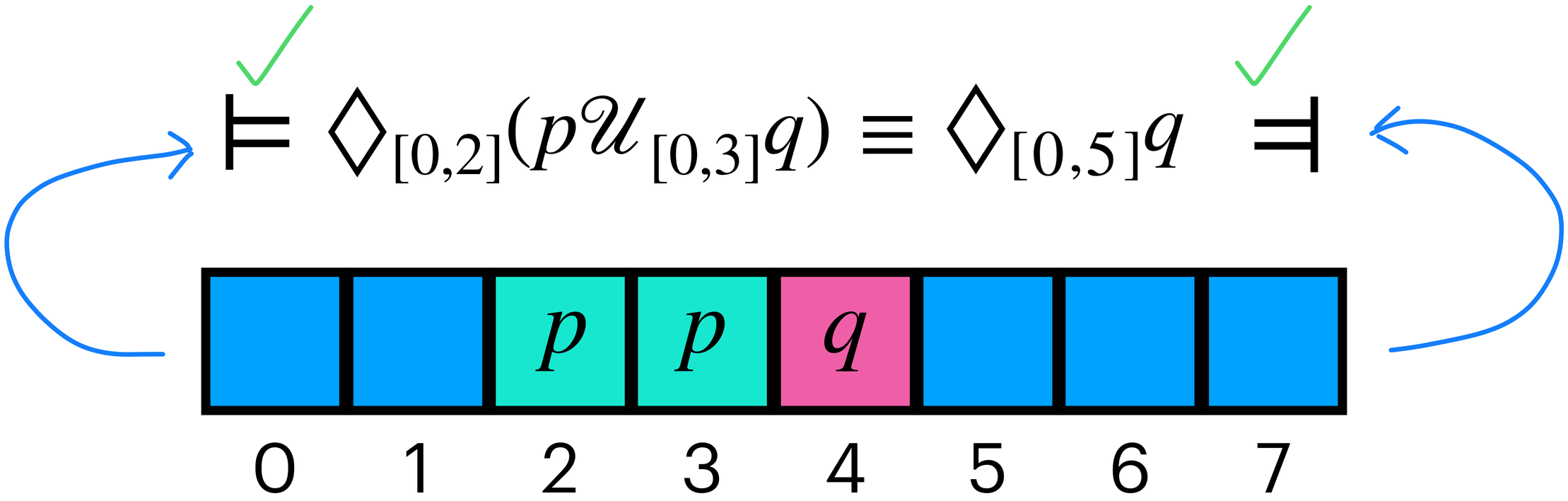
120

Does every LTL equiv. have a corollary in MLTL?



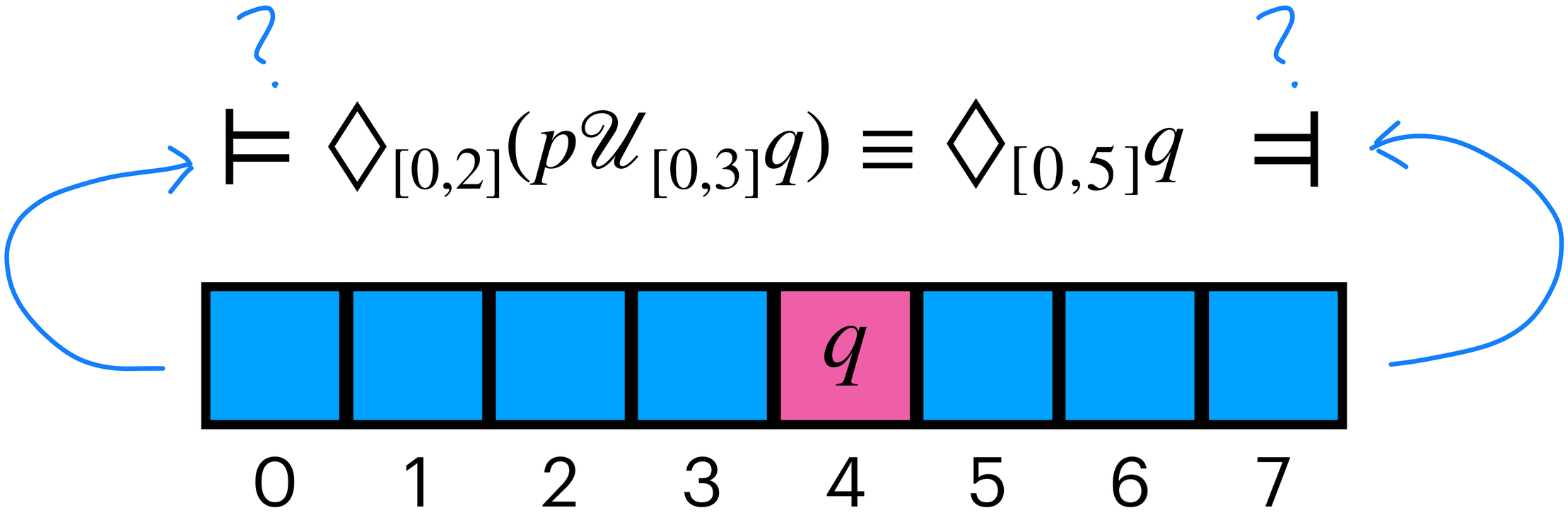
121

Does every LTL equiv. have a corollary in MLTL?



122

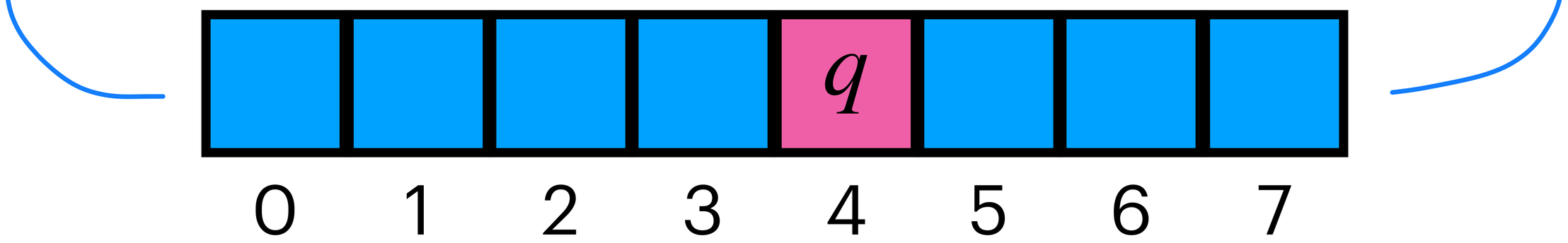
Does every LTL equiv. have a corollary in MLTL?



123

Does every LTL equiv. have a corollary in MLTL?

$$\not\models \Diamond_{[0,2]}(p \mathcal{U}_{[0,3]} q) \equiv \Diamond_{[0,5]} q \quad \not\exists$$



124

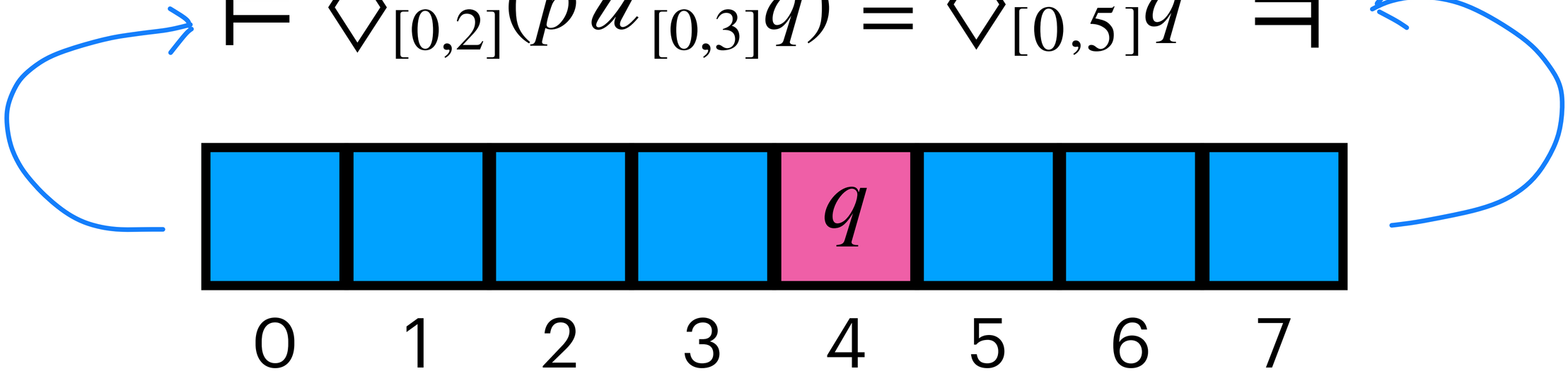
Does every LTL equiv. have a corollary in MLTL?

-NO!

X

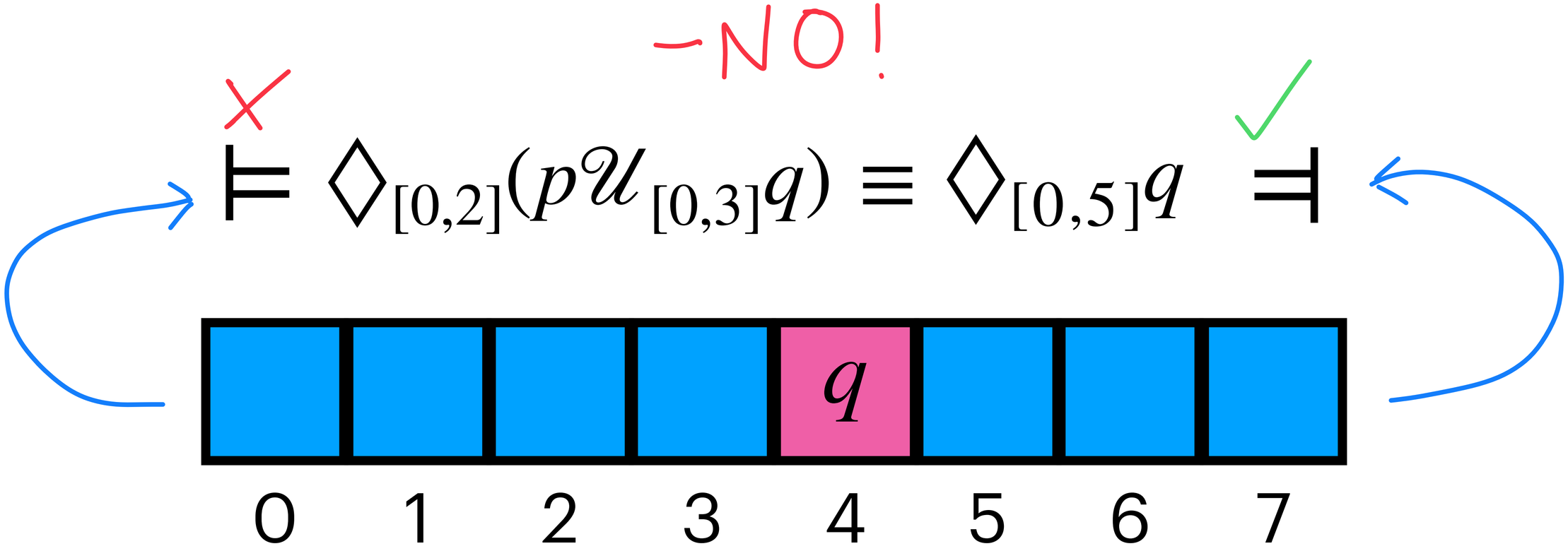
$$\models \Diamond_{[0,2]}(p \mathcal{U}_{[0,3]} q) \equiv \Diamond_{[0,5]} q \quad \nRightarrow$$

✓



125

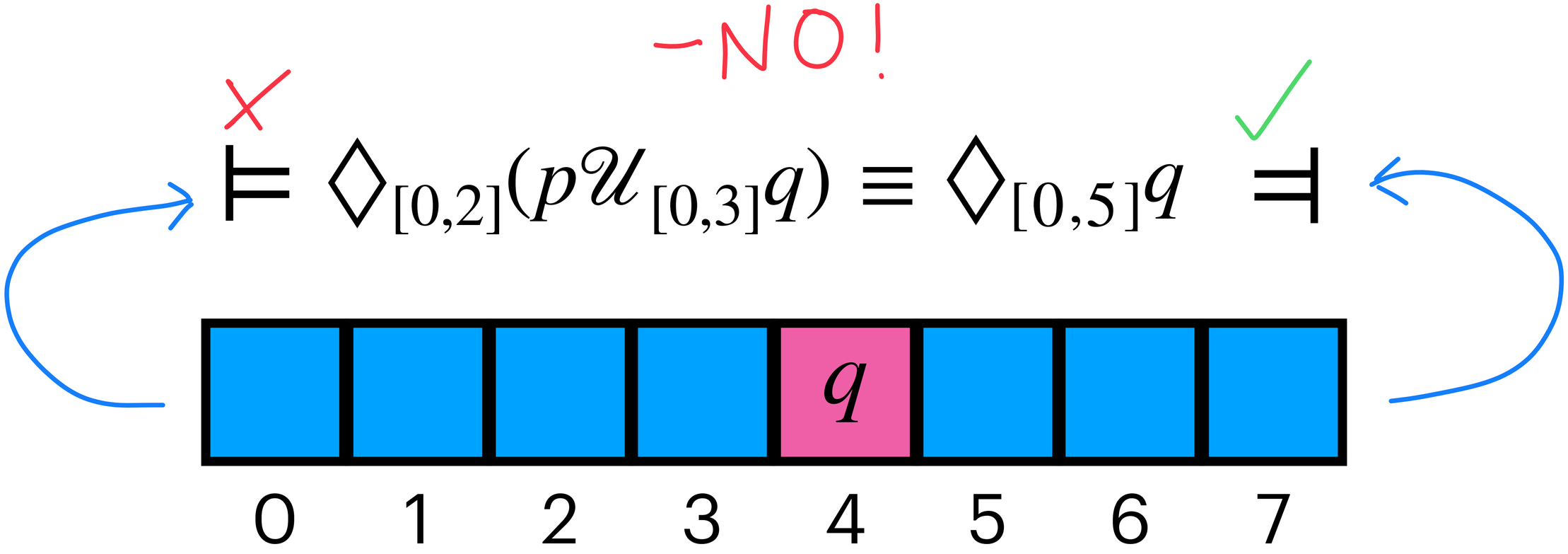
Does every LTL equiv. have a corollary in MLTL?



MLTL places stricter constraints on the set of satisfying traces (compared to LTL)

126

Does every LTL equiv. have a corollary in MLTL?



MLTL places stricter constraints on the set of satisfying traces (compared to LTL)

127

Our work presents:

- Encodings for “intractable” specifications using **implied domain constraints** in MLTL, providing **memory and time guarantees**.
- Set of **MLTL rewrite rules**, proofs of **correctness**, and proofs of **encoding size reduction** for each.

Future Work:

- More rewrite rules, tighter derivation of memory savings.
- Algorithm for finding minimal-sized MLTL encoding
- Quantifier-elimination

Special thanks to the NASA Lunar Gateway Vehicle System Manager team for valuable insight.

https://r2u2.temporallogic.org/

128