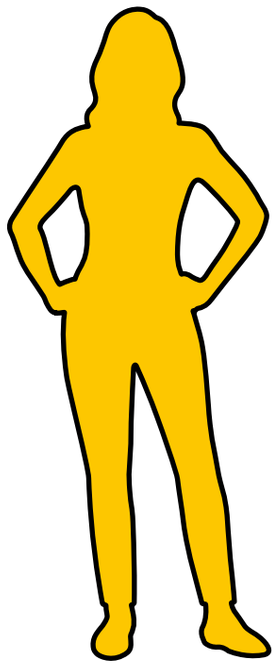
Dynamic Set Reasoning

SPECIFYING AND OPTIMIZING MONITOR ENCODINGS

Chris Johannsen

IOWA STATE UNIVERSITY

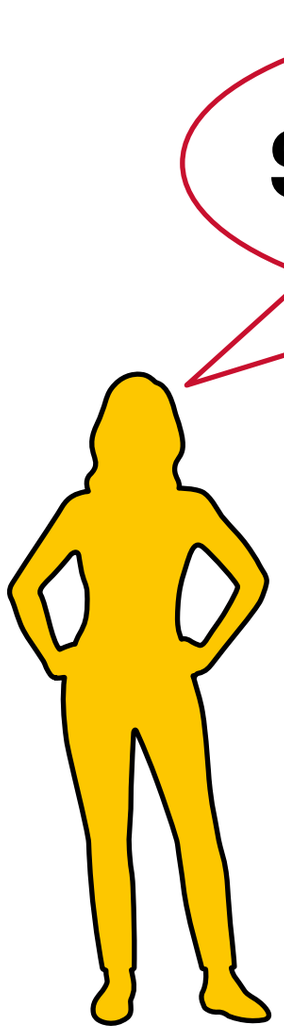
Sam the Satellite Engineer



Sam

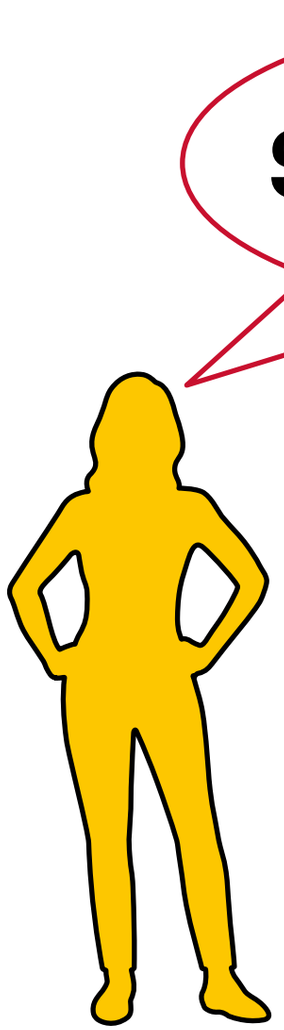
[2]

Sam the Satellite Engineer



[2]

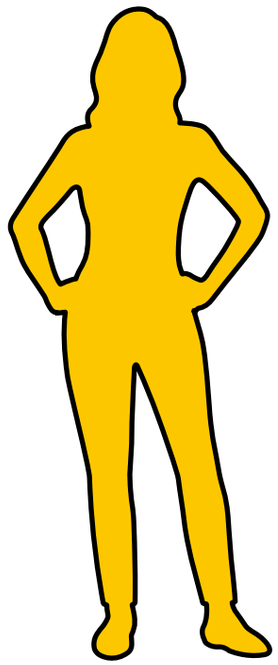
Sam the Satellite Engineer



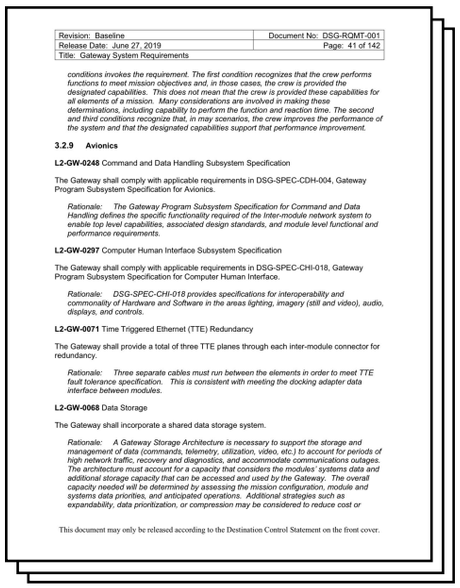
[2]

How to detect and diagnose “stuff”?

Sam the Satellite Engineer



Sam



Revision: Baseline	Document No: DSG-RQMT-001
Release Date: June 27, 2019	Page: 41 of 142
Title: Gateway System Requirements	

conditions invokes the requirement. The first condition recognizes that the crew performs functions to meet mission objectives and, in those cases, the crew is provided the designated capabilities. This does not mean that the crew is provided these capabilities for all elements of a mission. Many considerations are involved in making these determinations, including capability to perform the function and reaction time. The second and third conditions recognize that, in may scenarios, the crew improves the performance of the system and that the designated capabilities support that performance improvement.

3.2.9 Avionics

L2-GW-0248 Command and Data Handling Subsystem Specification

The Gateway shall comply with applicable requirements in DSG-SPEC-CDH-004, Gateway Program Subsystem Specification for Avionics.

Rationale: The Gateway Program Subsystem Specification for Command and Data Handling defines the specific functionality required of the Inter-module network system to enable top level capabilities, associated design standards, and module level functional and performance requirements.

L2-GW-0297 Computer Human Interface Subsystem Specification

The Gateway shall comply with applicable requirements in DSG-SPEC-CHI-018, Gateway Program Subsystem Specification for Computer Human Interface.

Rationale: DSG-SPEC-CHI-018 provides specifications for interoperability and commonality of Hardware and Software in the areas lighting, imagery (still and video), audio, displays, and controls.

L2-GW-0071 Time Triggered Ethernet (TTE) Redundancy

The Gateway shall provide a total of three TTE planes through each inter-module connector for redundancy.

Rationale: Three separate cables must run between the elements in order to meet TTE fault tolerance specification. This is consistent with meeting the docking adapter data interface between modules.

L2-GW-0068 Data Storage

The Gateway shall incorporate a shared data storage system.

Rationale: A Gateway Storage Architecture is necessary to support the storage and management of data (commands, telemetry, utilization, video, etc.) to account for periods of high network traffic, recovery and diagnostics, and accommodate communications outages. The architecture must account for a capacity that considers the modules' systems data and additional storage capacity that can be accessed and used by the Gateway. The overall capacity needed will be determined by assessing the mission configuration, module and systems data priorities, and anticipated operations. Additional strategies such as expandability, data prioritization, or compression may be considered to reduce cost or

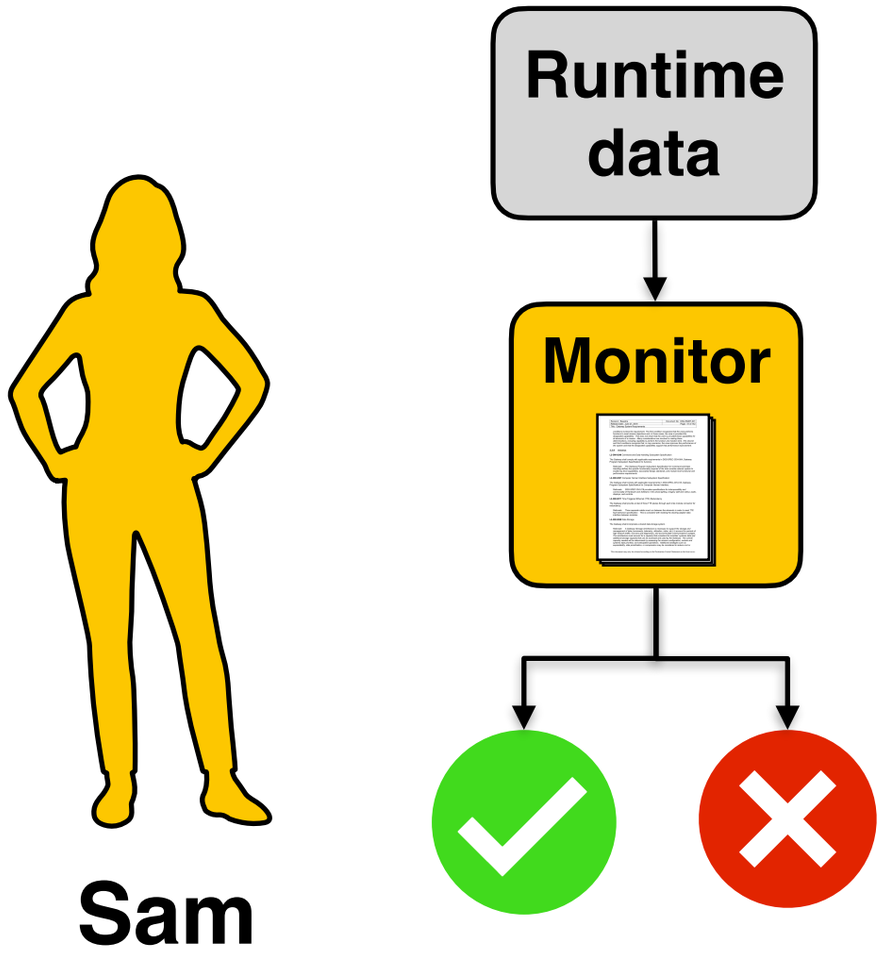
This document may only be released according to the Destination Control Statement on the front cover.

[4]

[2]

How to detect and diagnose “stuff”?

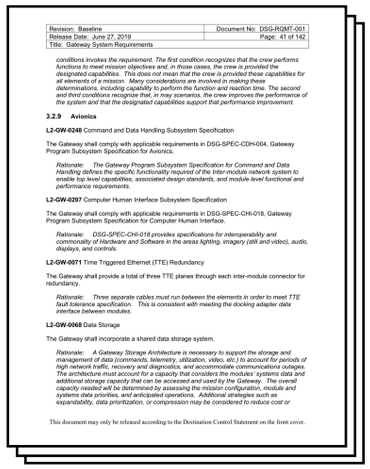
Sam the Satellite Engineer



[2]

How to detect and diagnose “stuff”?

How does Sam implement a monitor?



Revision: Baseline | Document No: DSG-RQMT-001
Release Date: June 27, 2019 | Page: 41 of 142
Title: Gateway System Requirements

conditions invokes the requirement. The first condition recognizes that the crew performs functions to meet mission objectives and, in those cases, the crew is provided the designated capabilities. This does not mean that the crew is provided these capabilities for all elements of a mission. Many considerations are involved in making these determinations, including capability to perform the function and reaction time. The second and third conditions recognize that, in may scenarios, the crew improves the performance of the system and that the designated capabilities support that performance improvement.

3.2.9 Avionics

L2-GW-0248 Command and Data Handling Subsystem Specification

The Gateway shall comply with applicable requirements in DSG-SPEC-CDH-004, Gateway Program Subsystem Specification for Avionics.

Rationale: The Gateway Program Subsystem Specification for Command and Data Handling defines the specific functionality required of the inter-module network system to enable top level capabilities, associated design standards, and module level functional and performance requirements.

L2-GW-0297 Computer Human Interface Subsystem Specification

The Gateway shall comply with applicable requirements in DSG-SPEC-CHI-018, Gateway Program Subsystem Specification for Computer Human Interface.

Rationale: DSG-SPEC-CHI-018 provides specifications for interoperability and commonality of Hardware and Software in the areas lighting, imagery (still and video), audio, displays, and controls.

L2-GW-0071 Time Triggered Ethernet (TTE) Redundancy

The Gateway shall provide a total of three TTE planes through each inter-module connector for redundancy.

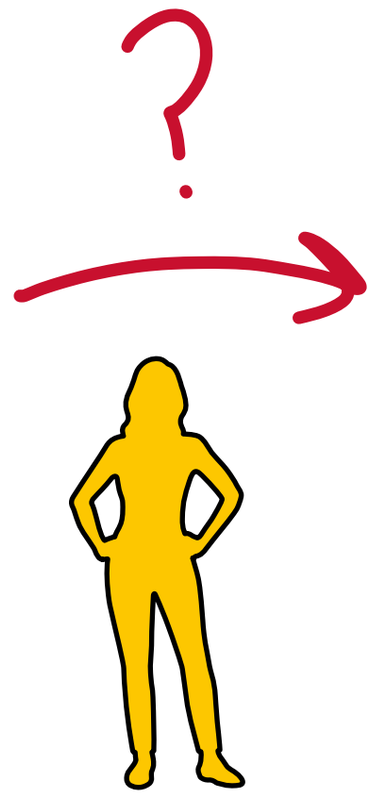
Rationale: Three separate cables must run between the elements in order to meet TTE fault tolerance specification. This is consistent with meeting the docking adapter data interface between modules.

L2-GW-0068 Data Storage

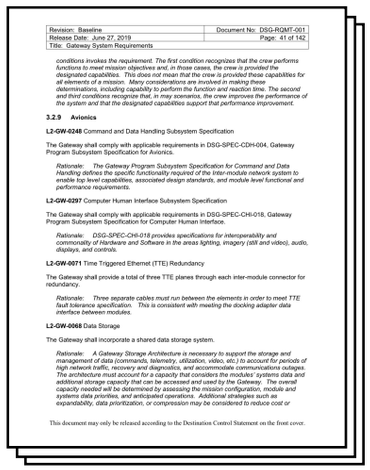
The Gateway shall incorporate a shared data storage system.

Rationale: A Gateway Storage Architecture is necessary to support the storage and management of data (commands, telemetry, utilization, video, etc.) to account for periods of high network traffic, recovery and diagnostics, and accommodate communications outages. The architecture must account for a capacity that considers the modules' systems data and additional storage capacity that can be accessed and used by the Gateway. The overall capacity needed will be determined by assessing the mission configuration, module and systems data priorities, and anticipated operations. Additional strategies such as expandability, data prioritization, or compression may be considered to reduce cost or

This document may only be released according to the Destination Control Statement on the front cover.



How does Sam implement a monitor?



Revision: Baseline | Document No: DSG-RQMT-001
Release Date: June 27, 2019 | Page: 41 of 142
Title: Gateway System Requirements

conditions invokes the requirement. The first condition recognizes that the crew performs functions to meet mission objectives and; in those cases, the crew is provided the designated capabilities. This does not mean that the crew is provided these capabilities for all elements of a mission. Many considerations are involved in making these determinations, including capability to perform the function and reaction time. The second and third conditions recognize that, in may scenarios, the crew improves the performance of the system and that the designated capabilities support that performance improvement.

3.2.9 Avionics

L2-GW-0248 Command and Data Handling Subsystem Specification

The Gateway shall comply with applicable requirements in DSG-SPEC-CDH-004, Gateway Program Subsystem Specification for Avionics.

Rationale: The Gateway Program Subsystem Specification for Command and Data Handling defines the specific functionality required of the inter-module network system to enable top level capabilities, associated design standards, and module level functional and performance requirements.

L2-GW-0297 Computer Human Interface Subsystem Specification

The Gateway shall comply with applicable requirements in DSG-SPEC-CHI-018, Gateway Program Subsystem Specification for Computer Human Interface.

Rationale: DSG-SPEC-CHI-018 provides specifications for interoperability and commonality of Hardware and Software in the areas lighting, imagery (still and video), audio, displays, and controls.

L2-GW-0071 Time Triggered Ethernet (TTE) Redundancy

The Gateway shall provide a total of three TTE planes through each inter-module connector for redundancy.

Rationale: Three separate cables must run between the elements in order to meet TTE fault tolerance specification. This is consistent with meeting the docking adapter data interface between modules.

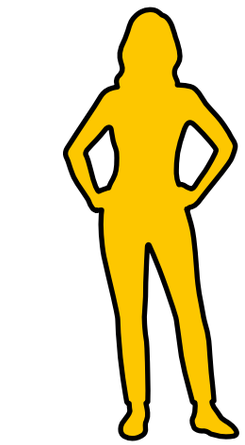
L2-GW-0068 Data Storage

The Gateway shall incorporate a shared data storage system.

Rationale: A Gateway Storage Architecture is necessary to support the storage and management of data (commands, telemetry, utilization, video, etc.) to account for periods of high network traffic, recovery and diagnostics, and accommodate communications outages. The architecture must account for a capacity that considers the modules' systems data and additional storage capacity that can be accessed and used by the Gateway. The overall capacity needed will be determined by assessing the mission configuration, module and systems data priorities, and anticipated operations. Additional strategies such as expandability, data prioritization, or compression may be considered to reduce cost or

This document may only be released according to the Destination Control Statement on the front cover.

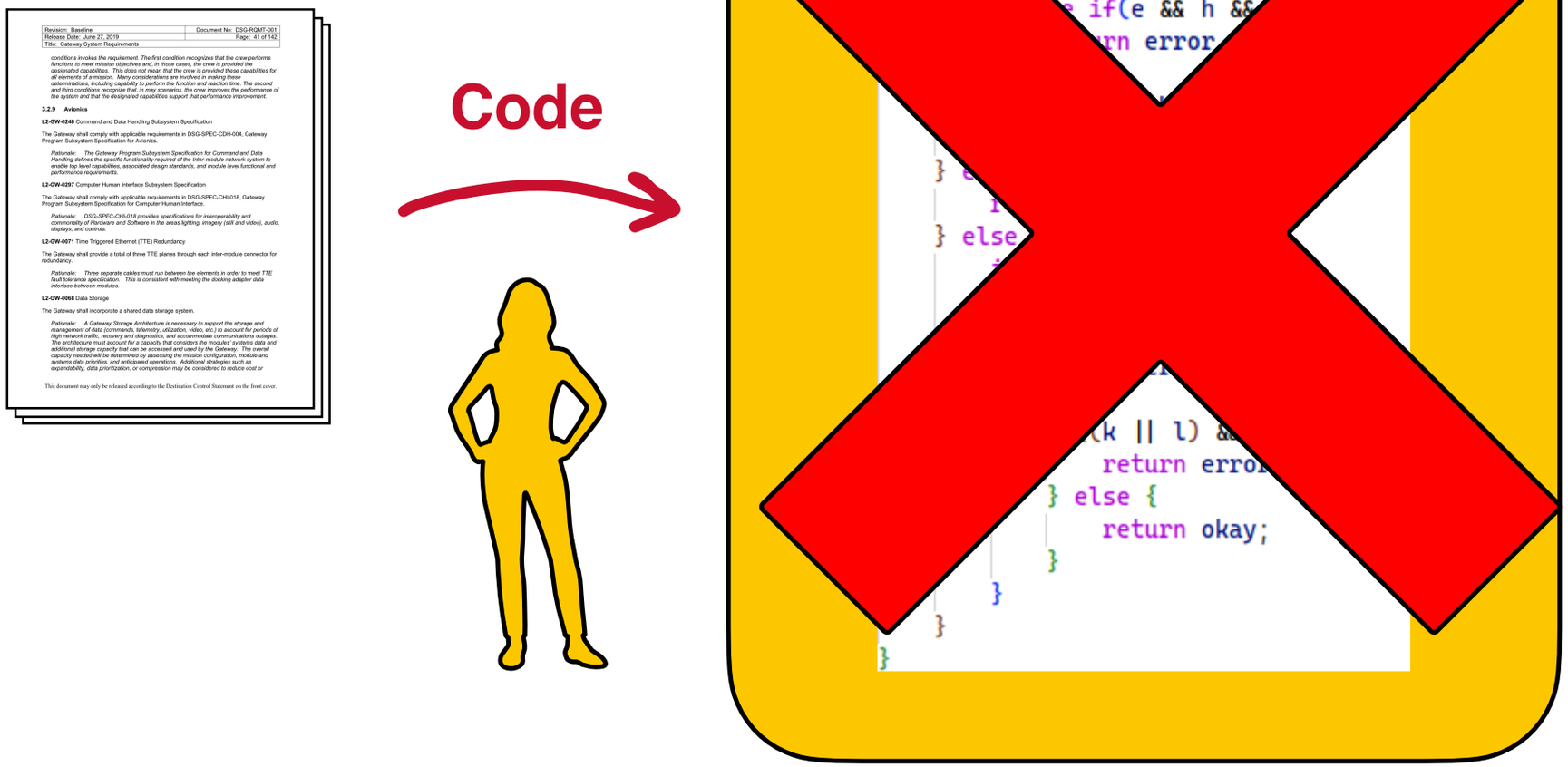
Code



Monitor

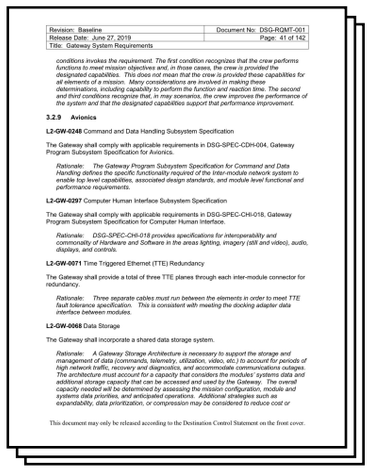
```
if(a && b) {
    if(!c || d) {
        return error_0;
    } else if(c || d) {
        if (e && f && g) {
            return error_1;
        } else if(e && h && g) {
            return error_2;
        } else {
            return okay;
        }
    } else if(d) {
        return error_1;
    } else {
        if(i && !j) {
            return error_4;
        } else if(!i && j) {
            return error_5;
        } else {
            if((k || l) && (k == m)) {
                return error_6;
            } else {
                return okay;
            }
        }
    }
}
```

How does Sam implement a monitor?

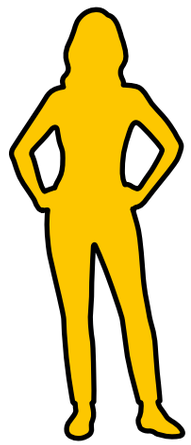


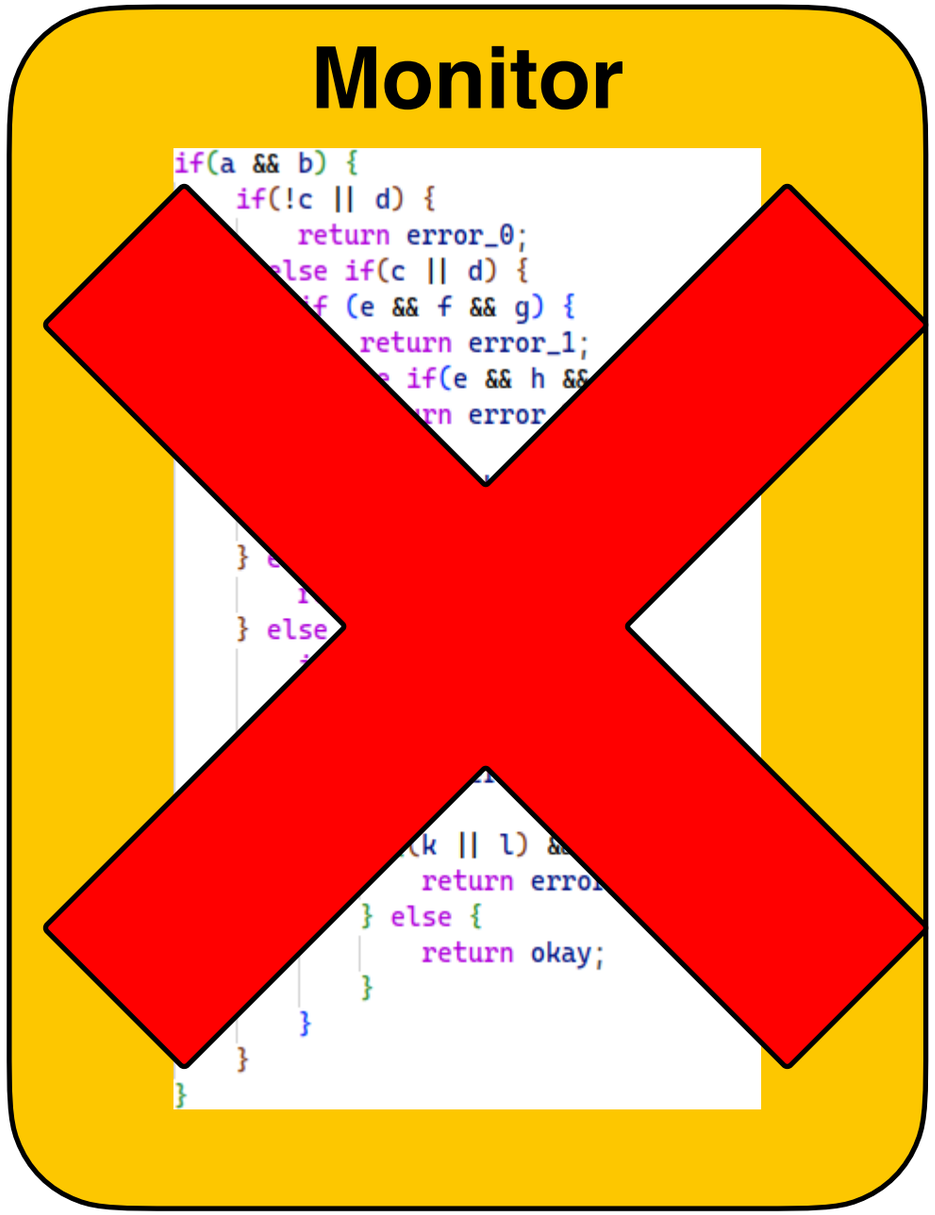
How does Sam implement a monitor?

- Re-certification
- Opaque

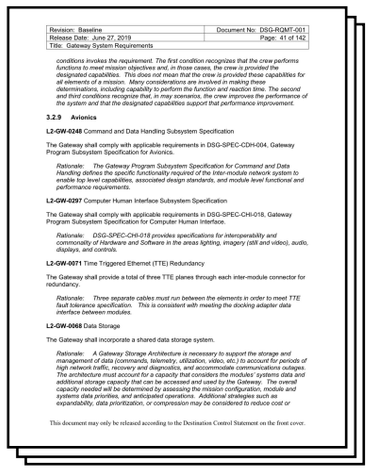


Code





Runtime Verification (RV) Approach


Revision: Baseline | Document No: DSG-RQMT-001
Release Date: June 27, 2019 | Page: 41 of 142
Title: Gateway System Requirements

conditions invokes the requirement. The first condition recognizes that the crew performs functions to meet mission objectives and, in those cases, the crew is provided the designated capabilities. This does not mean that the crew is provided these capabilities for all elements of a mission. Many considerations are involved in making these determinations, including capability to perform the function and reaction time. The second and third conditions recognize that, in may scenarios, the crew improves the performance of the system and that the designated capabilities support that performance improvement.

3.2.9 Avionics

L2-GW-0248 Command and Data Handling Subsystem Specification

The Gateway shall comply with applicable requirements in DSG-SPEC-CDH-004, Gateway Program Subsystem Specification for Avionics.

Rationale: The Gateway Program Subsystem Specification for Command and Data Handling defines the specific functionality required of the inter-module network system to enable top level capabilities, associated design standards, and module level functional and performance requirements.

L2-GW-0297 Computer Human Interface Subsystem Specification

The Gateway shall comply with applicable requirements in DSG-SPEC-CHI-018, Gateway Program Subsystem Specification for Computer Human Interface.

Rationale: DSG-SPEC-CHI-018 provides specifications for interoperability and commonality of Hardware and Software in the areas lighting, imagery (still and video), audio, displays, and controls.

L2-GW-0071 Time Triggered Ethernet (TTE) Redundancy

The Gateway shall provide a total of three TTE planes through each inter-module connector for redundancy.

Rationale: Three separate cables must run between the elements in order to meet TTE fault tolerance specification. This is consistent with meeting the docking adapter data interface between modules.

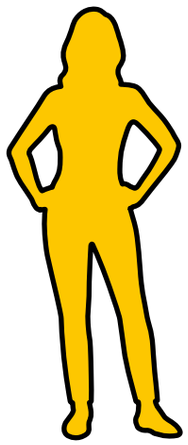
L2-GW-0068 Data Storage

The Gateway shall incorporate a shared data storage system.

Rationale: A Gateway Storage Architecture is necessary to support the storage and management of data (commands, telemetry, utilization, video, etc.) to account for periods of high network traffic, recovery and diagnostics, and accommodate communications outages. The architecture must account for a capacity that considers the modules' systems data and additional storage capacity that can be accessed and used by the Gateway. The overall capacity needed will be determined by assessing the mission configuration, module and systems data priorities, and anticipated operations. Additional strategies such as expandability, data prioritization, or compression may be considered to reduce cost or

This document may only be released according to the Destination Control Statement on the front cover.

Formalize

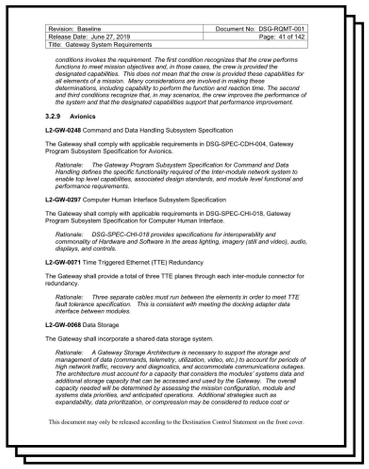


- Temporal Logic
- Automata
- LOLA [5]

Generate

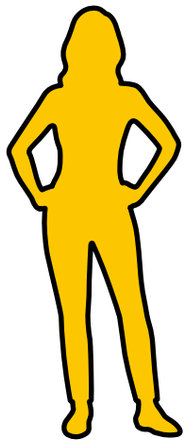
Monitor

Runtime Verification (RV) Approach



What to use?

Formalize



- Temporal Logic
- Automata
- LOLA [5]

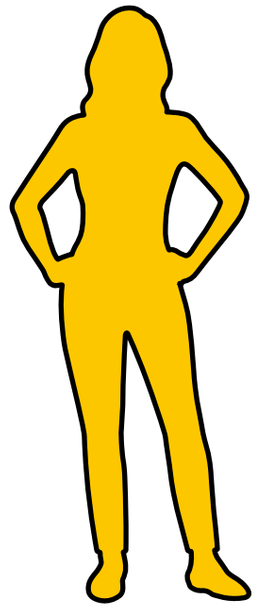
Generate

Monitor

Which RV tool to use?

Sam's constraints (**embedded, real-time, safety-critical**):

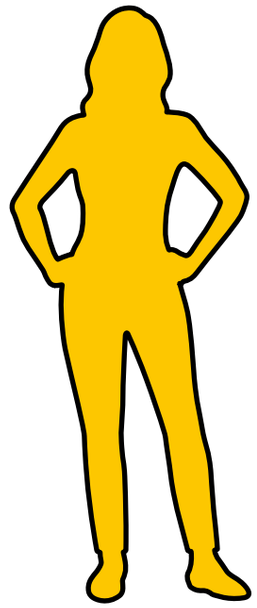
- **Realizable**
- **Responsive**
- **Unobtrusive**



Which RV tool to use?

Sam's constraints (**embedded, real-time, safety-critical**):

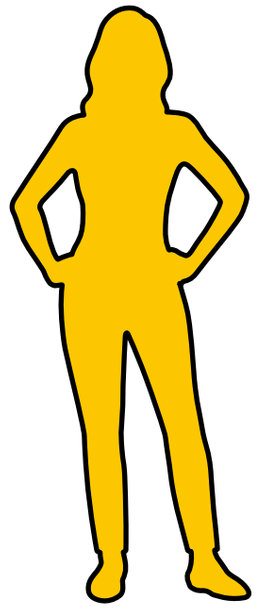
- **Realizable**
  - Use **bounded, little** resources
  - Adapt to new specifications **without re-compilation**
  - **Expressive, intuitive** specification language
- **Responsive**
- **Unobtrusive**



Which RV tool to use?

Sam's constraints (**embedded, real-time, safety-critical**):

- **Realizable**
  - Use **bounded, little** resources
  - Adapt to new specifications **without re-compilation**
  - **Expressive, intuitive** specification language
- **Responsive**
  - Runs continuously
  - Provides results **as soon as available**
- **Unobtrusive**

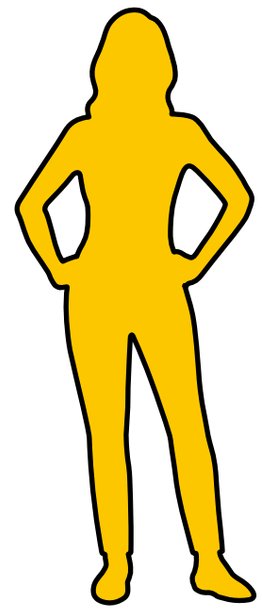


Which RV tool to use?

Sam's constraints (**embedded, real-time, safety-critical**):

- **Realizable**
  - Use **bounded, little** resources
  - Adapt to new specifications **without re-compilation**
  - **Expressive, intuitive** specification language

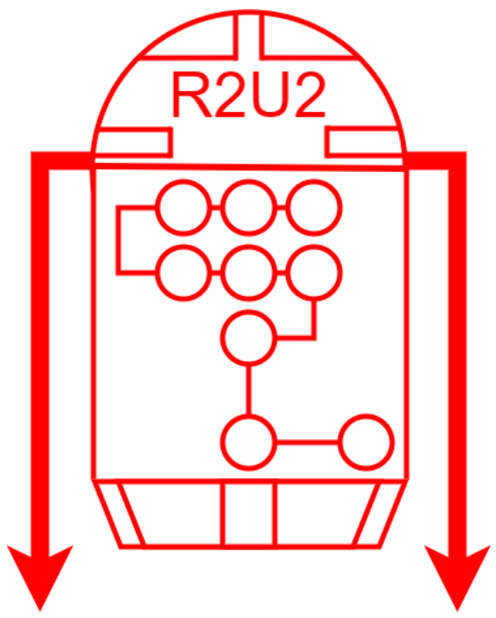
- **Responsive**
  - Runs continuously
  - Provides results **as soon as available**
- **Unobtrusive**
  - Does not impact **certifiability/functionality/behavior**
  - Black box



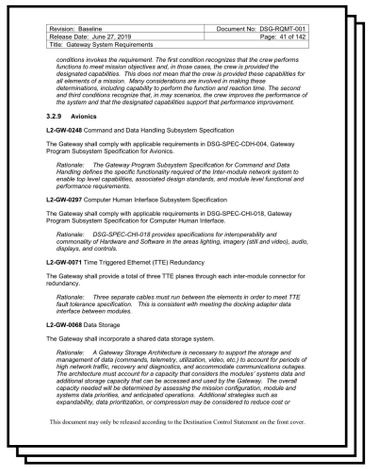
Which RV tool to use?

Sam's constraints (**embedded, real-time, safety-critical**):

- **Realizable**
- **Responsive**
- **Unobtrusive**
- **Unit**



R2U2 Approach



Revision: Baseline | Document No: DSG-RQMT-001
Release Date: June 27, 2019 | Page: 41 of 142
Title: Gateway System Requirements

conditions invokes the requirement. The first condition recognizes that the crew performs functions to meet mission objectives and, in those cases, the crew is provided the designated capabilities. This does not mean that the crew is provided these capabilities for all elements of a mission. Many considerations are involved in making these determinations, including capability to perform the function and reaction time. The second and third conditions recognize that, in may scenarios, the crew improves the performance of the system and that the designated capabilities support that performance improvement.

3.2.9 Avionics

L2-GW-0248 Command and Data Handling Subsystem Specification

The Gateway shall comply with applicable requirements in DSG-SPEC-CDH-004, Gateway Program Subsystem Specification for Avionics.

Rationale: The Gateway Program Subsystem Specification for Command and Data Handling defines the specific functionality required of the inter-module network system to enable top level capabilities, associated design standards, and module level functional and performance requirements.

L2-GW-0297 Computer Human Interface Subsystem Specification

The Gateway shall comply with applicable requirements in DSG-SPEC-CHI-018, Gateway Program Subsystem Specification for Computer Human Interface.

Rationale: DSG-SPEC-CHI-018 provides specifications for interoperability and commonality of Hardware and Software in the areas lighting, imagery (still and video), audio, displays, and controls.

L2-GW-0071 Time Triggered Ethernet (TTE) Redundancy

The Gateway shall provide a total of three TTE planes through each inter-module connector for redundancy.

Rationale: Three separate cables must run between the elements in order to meet TTE fault tolerance specification. This is consistent with meeting the docking adapter data interface between modules.

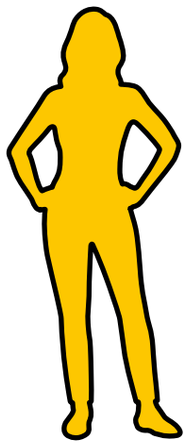
L2-GW-0068 Data Storage

The Gateway shall incorporate a shared data storage system.

Rationale: A Gateway Storage Architecture is necessary to support the storage and management of data (commands, telemetry, utilization, video, etc.) to account for periods of high network traffic, recovery and diagnostics, and accommodate communications outages. The architecture must account for a capacity that considers the modules' systems data and additional storage capacity that can be accessed and used by the Gateway. The overall capacity needed will be determined by assessing the mission configuration, module and systems data priorities, and anticipated operations. Additional strategies such as expandability, data prioritization, or compression may be considered to reduce cost or

This document may only be released according to the Destination Control Statement on the front cover.

Formalize

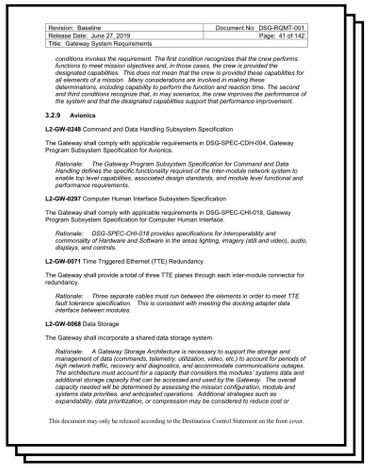


- Temporal Logic
- Automata
- LOLA [5]

Generate

Monitor

R2U2 Approach



Revision: Baseline | Document No: DSG-RQMT-001
Release Date: June 27, 2019 | Page: 41 of 142
Title: Gateway System Requirements

conditions invokes the requirement. The first condition recognizes that the crew performs functions to meet mission objectives and, in those cases, the crew is provided the designated capabilities. This does not mean that the crew is provided these capabilities for all elements of a mission. Many considerations are involved in making these determinations, including capability to perform the function and reaction time. The second and third conditions recognize that, in may scenarios, the crew improves the performance of the system and that the designated capabilities support that performance improvement.

3.2.9 Avionics

L2-GW-0248 Command and Data Handling Subsystem Specification

The Gateway shall comply with applicable requirements in DSG-SPEC-CDH-004, Gateway Program Subsystem Specification for Avionics.

Rationale: The Gateway Program Subsystem Specification for Command and Data Handling defines the specific functionality required of the inter-module network system to enable top level capabilities, associated design standards, and module level functional and performance requirements.

L2-GW-0297 Computer Human Interface Subsystem Specification

The Gateway shall comply with applicable requirements in DSG-SPEC-CHI-018, Gateway Program Subsystem Specification for Computer Human Interface.

Rationale: DSG-SPEC-CHI-018 provides specifications for interoperability and commonality of Hardware and Software in the areas lighting, imagery (still and video), audio, displays, and controls.

L2-GW-0071 Time Triggered Ethernet (TTE) Redundancy

The Gateway shall provide a total of three TTE planes through each inter-module connector for redundancy.

Rationale: Three separate cables must run between the elements in order to meet TTE fault tolerance specification. This is consistent with meeting the docking adapter data interface between modules.

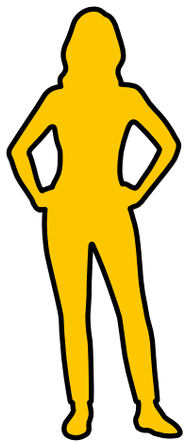
L2-GW-0068 Data Storage

The Gateway shall incorporate a shared data storage system.

Rationale: A Gateway Storage Architecture is necessary to support the storage and management of data (commands, telemetry, utilization, video, etc.) to account for periods of high network traffic, recovery and diagnostics, and accommodate communications outages. The architecture must account for a capacity that considers the modules' systems data and additional storage capacity that can be accessed and used by the Gateway. The overall capacity needed will be determined by assessing the mission configuration, module and systems data priorities, and anticipated operations. Additional strategies such as expandability, data prioritization, or compression may be considered to reduce cost or

This document may only be released according to the Destination Control Statement on the front cover.

Formalize

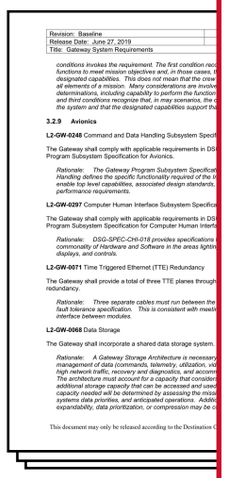


Mission-time LTL

Generate

Monitor

R2U2 Approach


Revision: Baseline
Release Date: June 27, 2019
Title: Gateway System Requirements

conditions invokes the requirement. The first condition rec
functions to meet mission objectives and, in those cases, t
designated capabilities. This does not mean that the crew
all elements of a mission. Many considerations are involve
determinations, including capability to perform the function
and third conditions recognize that, in may scenarios, the c
the system and that the designated capabilities support th

3.2.9 Avionics

L2-GW-0248 Command and Data Handling Subsystem Specifi

The Gateway shall comply with applicable requirements in DS
Program Subsystem Specification for Avionics.

*Rationale: The Gateway Program Subsystem Specificat
Handling defines the specific functionality required of the
enable top level capabilities, associated design standards,
performance requirements.*

L2-GW-0297 Computer Human Interface Subsystem Specifica

The Gateway shall comply with applicable requirements in DS
Program Subsystem Specification for Computer Human Interfa

*Rationale: DSG-SPEC-CHI-018 provides specifications
commonality of Hardware and Software in the areas lighti
displays, and controls.*

L2-GW-0071 Time Triggered Ethernet (TTE) Redundancy

The Gateway shall provide a total of three TTE planes through
redundancy.

*Rationale: Three separate cables must run between the
fault tolerance specification. This is consistent with missi
interface between modules.*

L2-GW-0068 Data Storage

The Gateway shall incorporate a shared data storage system.

*Rationale: A Gateway Storage Architecture is necessary
management of data (commands, telemetry, utilization, vi
high network traffic, recovery and diagnostics, and access
The architecture must account for a capacity that consider
additional storage capacity that can be accessed and used
capacity needed will be determined by assessing the missi
systems data priorities, and anticipated operations. Additio
expandability, data prioritization, or compression may be c*

This document may only be released according to the Destination C

Mission-time LTL

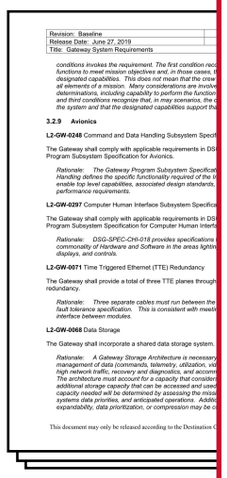
(MLTL) [16]

“If ‘task’ is in the scheduler, ‘task’ shall execute within 3 seconds ”

itor

$$(sched_{task}) \rightarrow (\Diamond_{[0,3]} exec_{task})$$

R2U2 Approach


Revision: Baseline
Release Date: June 27, 2019
Title: Gateway System Requirements

conditions invokes the requirement. The first condition rec
functions to meet mission objectives and, in those cases, t
designated capabilities. This does not mean that the crew
all elements of a mission. Many considerations are involve
determinations, including capability to perform the function
and third conditions recognize that, in may scenarios, the c
the system and that the designated capabilities support th

3.2.9 Avionics

L2-GW-0248 Command and Data Handling Subsystem Specifi

The Gateway shall comply with applicable requirements in DS
Program Subsystem Specification for Avionics.

*Rationale: The Gateway Program Subsystem Specificat
Handling defines the specific functionality required of the
enable top level capabilities, associated design standards,
performance requirements.*

L2-GW-0297 Computer Human Interface Subsystem Specifica

The Gateway shall comply with applicable requirements in DS
Program Subsystem Specification for Computer Human Interfa

*Rationale: DSG-SPEC-CHI-018 provides specifications
commonality of Hardware and Software in the areas lighti
displays, and controls.*

L2-GW-0071 Time Triggered Ethernet (TTE) Redundancy

The Gateway shall provide a total of three TTE planes through
redundancy.

*Rationale: Three separate cables must run between the
fault tolerance specification. This is consistent with maxim
interface between modules.*

L2-GW-0068 Data Storage

The Gateway shall incorporate a shared data storage system.

*Rationale: A Gateway Storage Architecture is necessary
management of data (commands, telemetry, utilization, vi
high network traffic, recovery and diagnostics, and access
The architecture must account for a capacity that consider
additional storage capacity that can be accessed and used
capacity needed will be determined by assessing the missi
systems data priorities, and anticipated operations. Additio
expandability, data prioritization, or compression may be c*

This document may only be released according to the Destination C

Mission-time LTL

(MLTL) [16]

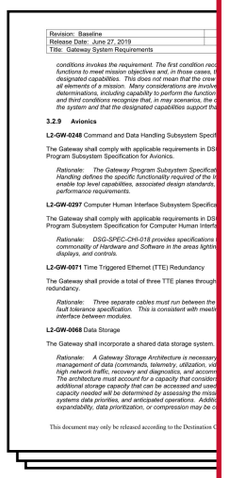
“If ‘task’ is in the scheduler, ‘task’ shall execute within 3 seconds ”

itor

$$(sched_{task}) \rightarrow (\Diamond_{[0,3]} exec_{task})$$

Propositions

R2U2 Approach


Revision: Baseline
Release Date: June 27, 2019
Title: Gateway System Requirements

conditions invokes the requirement. The first condition rec
functions to meet mission objectives and, in those cases, t
designated capabilities. This does not mean that the crew
all elements of a mission. Many considerations are involve
determinations, including capability to perform the function
and third conditions recognize that, in may scenarios, the c
the system and that the designated capabilities support th

3.2.9 Avionics

L2-GW-0248 Command and Data Handling Subsystem Specifi

The Gateway shall comply with applicable requirements in DS
Program Subsystem Specification for Avionics.

*Rationale: The Gateway Program Subsystem Specificat
Handling defines the specific functionality required of the
enable top level capabilities, associated design standards,
performance requirements.*

L2-GW-0297 Computer Human Interface Subsystem Specifica

The Gateway shall comply with applicable requirements in DS
Program Subsystem Specification for Computer Human Interfa

*Rationale: DSG-SPEC-CHI-018 provides specifications
commonality of Hardware and Software in the areas lighti
displays, and controls.*

L2-GW-0071 Time Triggered Ethernet (TTE) Redundancy

The Gateway shall provide a total of three TTE planes through
redundancy.

*Rationale: Three separate cables must run between the
fault tolerance specification. This is consistent with missi
interface between modules.*

L2-GW-0068 Data Storage

The Gateway shall incorporate a shared data storage system.

*Rationale: A Gateway Storage Architecture is necessary
management of data (commands, telemetry, utilization, vi
high network traffic, recovery and diagnostics, and access
The architecture must account for a capacity that consider
additional storage capacity that can be accessed and used
capacity needed will be determined by assessing the missi
systems data priorities, and anticipated operations. Addit
expandability, data prioritization, or compression may be c*

This document may only be released according to the Destination C

Mission-time LTL

(MLTL) [16]

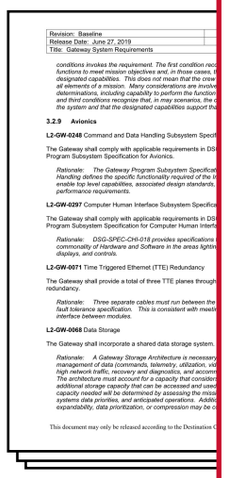
“If ‘task’ is in the scheduler, ‘task’ shall execute within 3 seconds ”

$$(sched_{task}) \rightarrow (\Diamond_{[0,3]} exec_{task})$$

Logical connectives ($\rightarrow$, $\neg$, $\wedge$, $\vee$)

itor

R2U2 Approach


Revision: Baseline
Release Date: June 27, 2019
Title: Gateway System Requirements

conditions invokes the requirement. The first condition rec
functions to meet mission objectives and, in those cases, t
designated capabilities. This does not mean that the crew
all elements of a mission. Many considerations are involve
determinations, including capability to perform the function
and third conditions recognize that, in may scenarios, the c
the system and that the designated capabilities support th

3.2.9 Avionics

L2-GW-0248 Command and Data Handling Subsystem Specifi

The Gateway shall comply with applicable requirements in DS
Program Subsystem Specification for Avionics.

*Rationale: The Gateway Program Subsystem Specificat
Handling defines the specific functionality required of the
enable top level capabilities, associated design standards,
performance requirements.*

L2-GW-0297 Computer Human Interface Subsystem Specifica

The Gateway shall comply with applicable requirements in DS
Program Subsystem Specification for Computer Human Interfa

*Rationale: DSG-SPEC-CHI-018 provides specifications
commonality of Hardware and Software in the areas lighti
displays, and controls.*

L2-GW-0071 Time Triggered Ethernet (TTE) Redundancy

The Gateway shall provide a total of three TTE planes through
redundancy.

*Rationale: Three separate cables must run between the
fault tolerance specification. This is consistent with missi
interface between modules.*

L2-GW-0068 Data Storage

The Gateway shall incorporate a shared data storage system.

*Rationale: A Gateway Storage Architecture is necessary
management of data (commands, telemetry, utilization, vi
high network traffic, recovery and diagnostics, and access
The architecture must account for a capacity that consider
additional storage capacity that can be accessed and used
capacity needed will be determined by assessing the missi
systems data priorities, and anticipated operations. Additio
expandability, data prioritization, or compression may be c*

This document may only be released according to the Destination C

Mission-time LTL

(MLTL) [16]

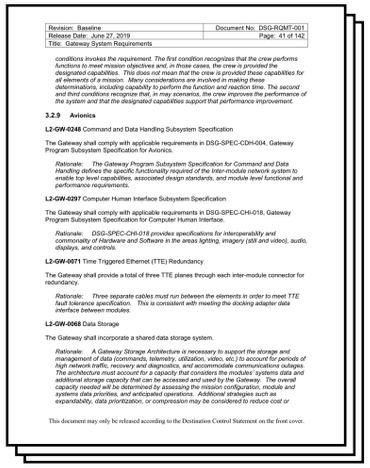
“If ‘task’ is in the scheduler, ‘task’ shall execute within 3 seconds ”

itor

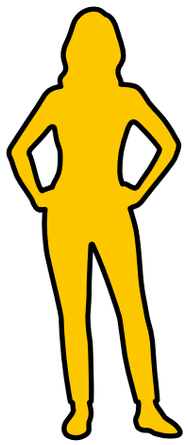
$$(sched_{task}) \rightarrow (\Diamond_{[0,3]} exec_{task})$$

Future-time, bounded ops

R2U2 Approach



Formalize

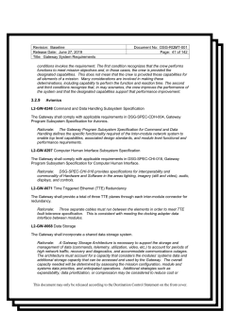


Mission-time LTL

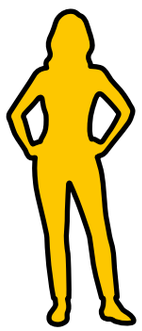
Generate

Monitor

R2U2 Approach



Formalize

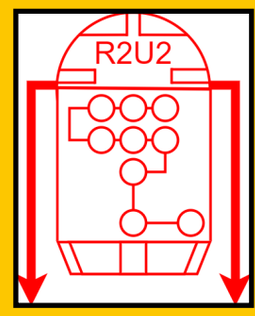


Mission-time LTL

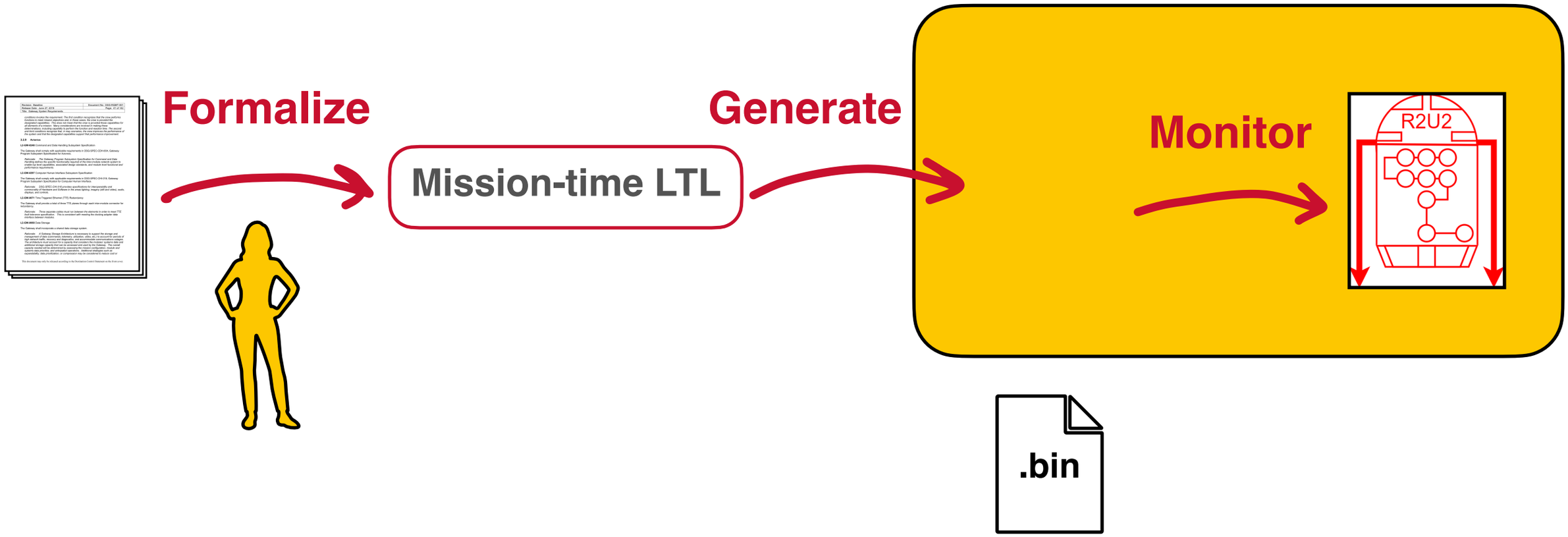
Generate

.bin

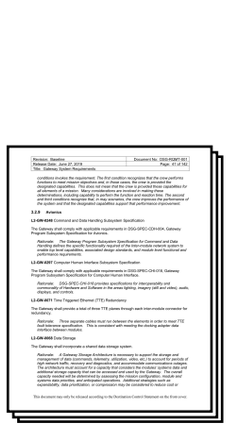
Monitor



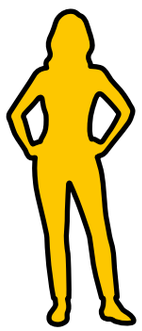
R2U2 Approach



R2U2 Approach



Formalize

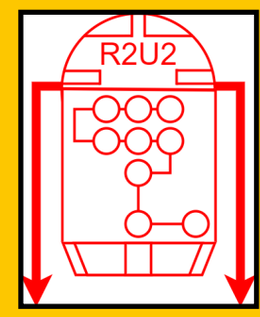


Mission-time LTL

Generate

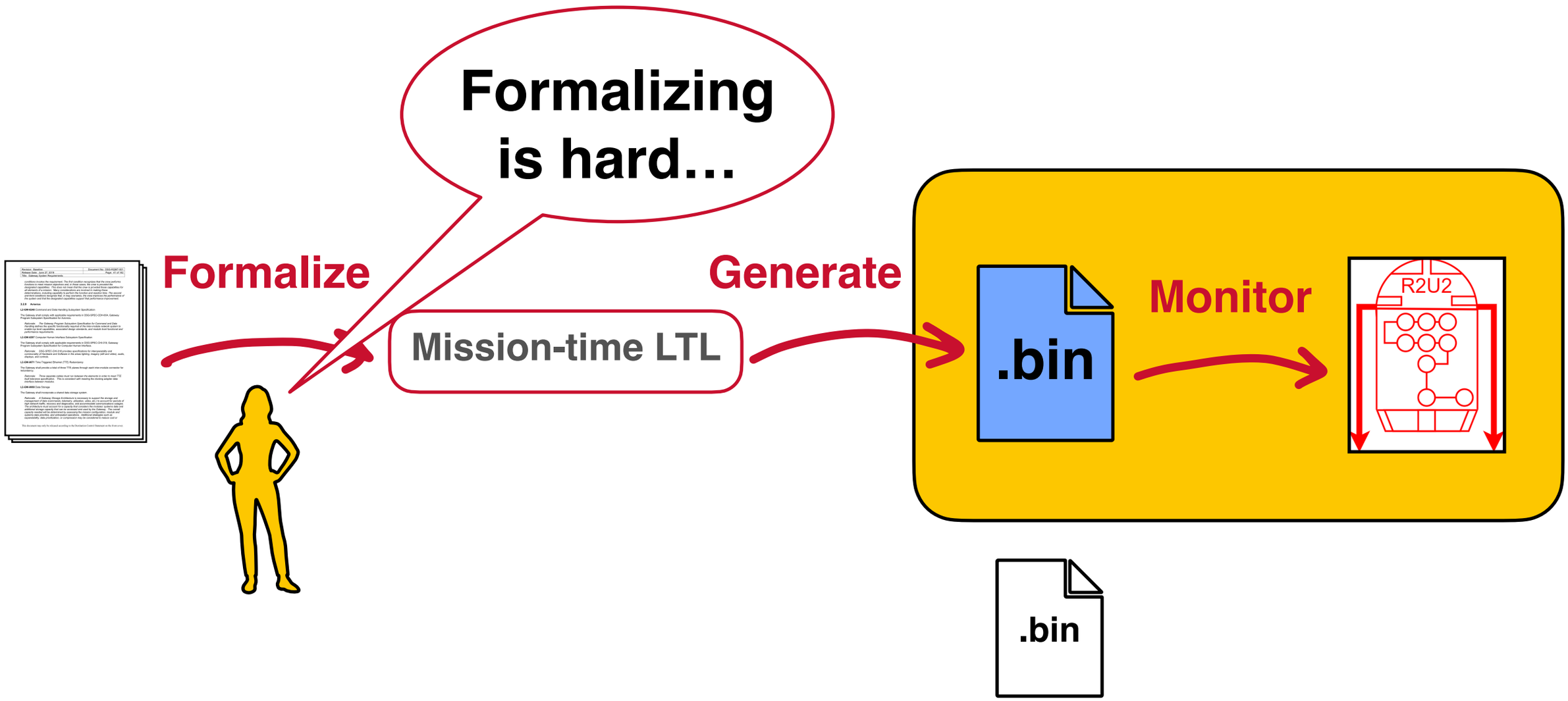
.bin

Monitor



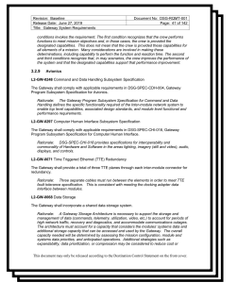
.bin

R2U2 Approach

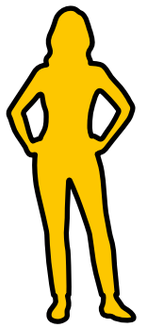


R2U2 + C2PO Approach

Configuration Compiler for Property Organization (C2PO) [17]



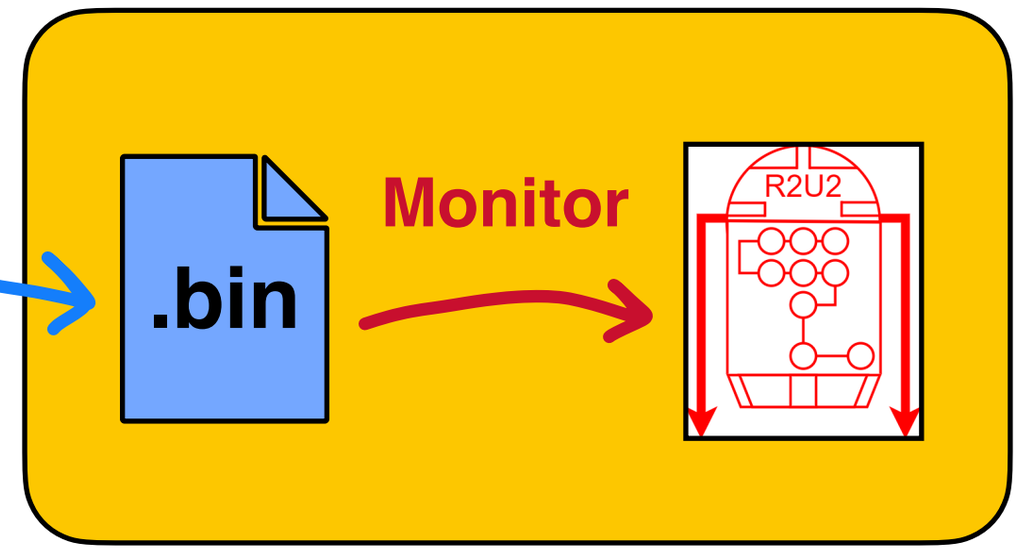
Formalize



```
STRUCT
    Task: {
        id: int;
        in_sched, exec: bool;
    };

INPUT
    task_id: int;
    task_in_sched, task_exec: bool;

DEFINE
    task := Task(
        task_id,
        task_in_sched,
        task_exec
    );

FTSPEC
    task.in_sched → F[0,3] task.exec;
```

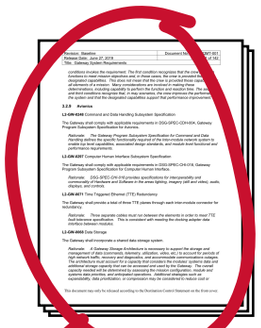
High-level language

Generate

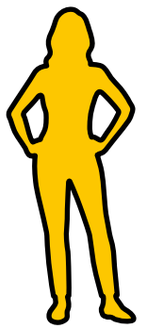


.bin

R2U2 + C2PO Approach



Formalize

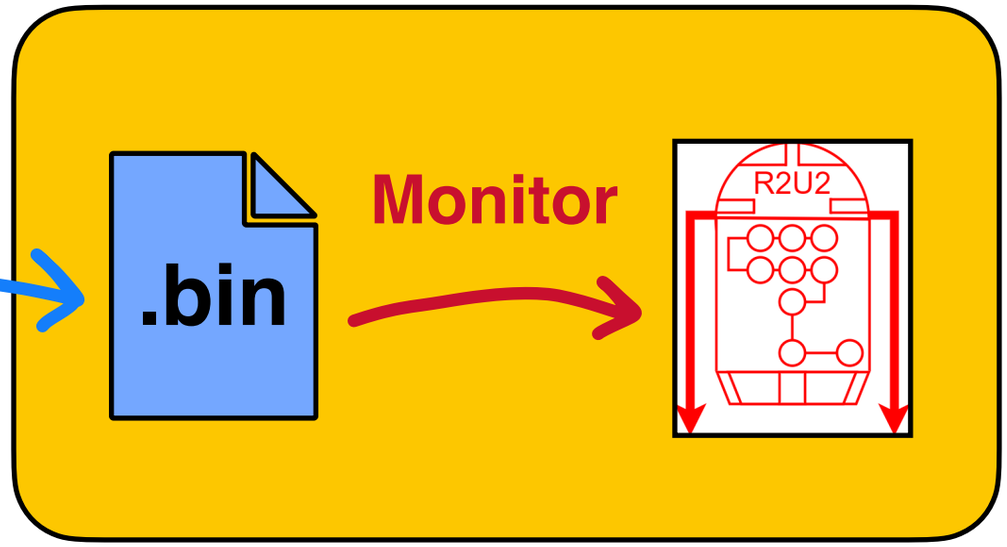


```
STRUCT
    Task: {
        id: int;
        in_sched, exec: bool;
    };

INPUT
    task_id: int;
    task_in_sched, task_exec: bool;

DEFINE
    task := Task(
        task_id,
        task_in_sched,
        task_exec
    );

FTSPEC
    task.in_sched → F[0,3] task.exec;
```

High-level language

Generate

Configuration Compiler for Property Organization (C2PO) [17]



.bin

“If ‘task’ is in the scheduler, ‘task’ shall execute within 3 seconds ”

C2PO Input Language [17]

“If ‘task’ is in the scheduler, ‘task’ shall execute within 3 seconds ”

```
STRUCT
    Task: {
        id: int;
        in_sched, exec: bool;
    };

INPUT
    task_id: int;
    task_in_sched, task_exec: bool;

DEFINE
    task := Task(
        task_id,
        task_in_sched,
        task_exec
    );

FTSPEC
    task.in_sched → F[0,3] task.exec;
```

C2PO Input Language [17]

“If ‘task’ is in the scheduler, ‘task’ shall execute within 3 seconds ”

```
STRUCT
    Task: {
        id: int;
        in_sched, exec: bool;
    };

INPUT
    task_id: int;
    task_in_sched, task_exec: bool;

DEFINE
    task := Task(
        task_id,
        task_in_sched,
        task_exec
    );

FTSPEC
    task.in_sched → F[0,3] task.exec;
```

C2PO Input Language [17]

“If ‘task’ is in the scheduler, ‘task’ shall execute within 3 seconds ”

```
STRUCT
    Task: {
        id: int;
        in_sched, exec: bool;
    };

INPUT
    task_id: int;
    task_in_sched, task_exec: bool;

DEFINE
    task := Task(
        task_id,
        task_in_sched,
        task_exec
    );

FTSPEC
    task.in_sched → F[0,3] task.exec;
```

C2PO Input Language [17]

“If ‘task’ is in the scheduler, ‘task’ shall execute within 3 seconds ”

```
STRUCT
    Task: {
        id: int;
        in_sched, exec: bool;
    };

INPUT
    task_id: int;
    task_in_sched, task_exec: bool;

DEFINE
    task := Task(
        task_id,
        task_in_sched,
        task_exec
    );

FTSPEC
    task.in_sched → F[0,3] task.exec;
```

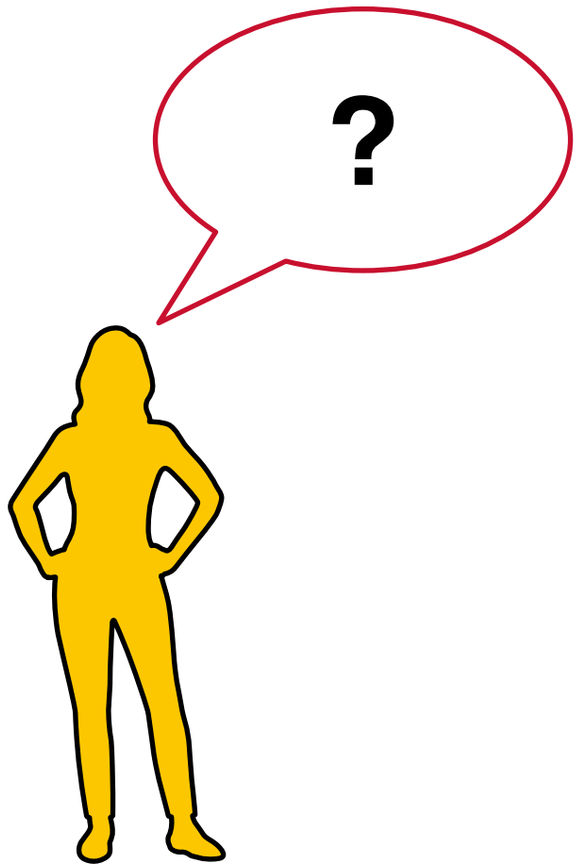
C2PO Input Language [17]

“If ‘task’ is in the scheduler, ‘task’ shall execute within 3 seconds ”

```
STRUCT
    Task: {
        id: int;
        in_sched, exec: bool;
    };

INPUT
    task_id: int;
    task_in_sched, task_exec: bool;

DEFINE
    task := Task(
        task_id,
        task_in_sched,
        task_exec
    );

FTSPEC
    task.in_sched → F[0,3] task.exec;
```

Sam's Specification Revisited

"If 'task' is in the scheduler, 'task' shall execute within 3 seconds "



Sam's Specification Revisited

Every task in the scheduler

"~~If 'task' is in the scheduler, 'task~~' shall execute within 3 seconds "

Sam's Specification Revisited

Every task in the scheduler

"~~If 'task' is in the scheduler, 'task~~" shall execute within 3 seconds "

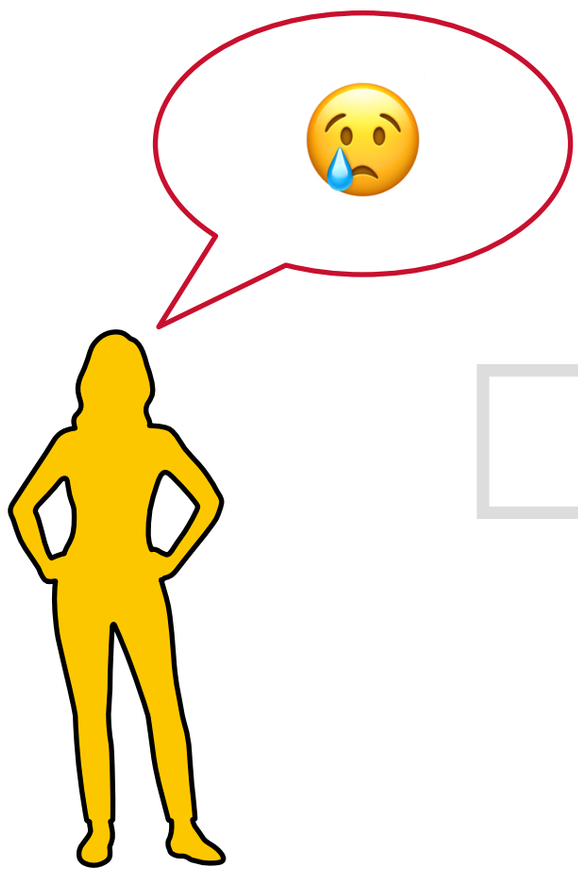
Formalize

$$\square \; \forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Sam's Specification Revisited

Every task in the scheduler

"~~If 'task' is in the scheduler, 'task~~' shall execute within 3 seconds "



Formalize

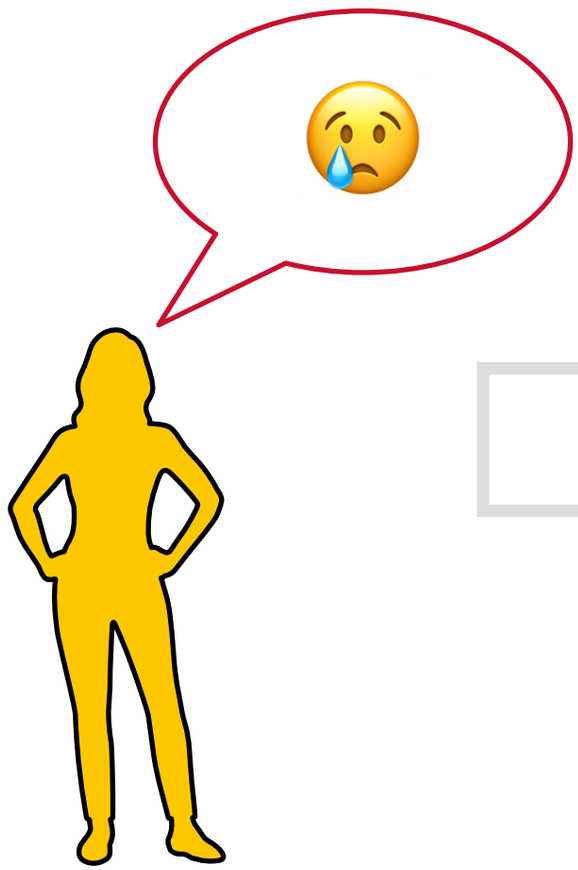
$$\Box \forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Can't use R2U2

Sam's Specification Revisited

Every task in the scheduler

"~~If 'task' is in the scheduler, 'task~~' shall execute within 3 seconds "



Formalize

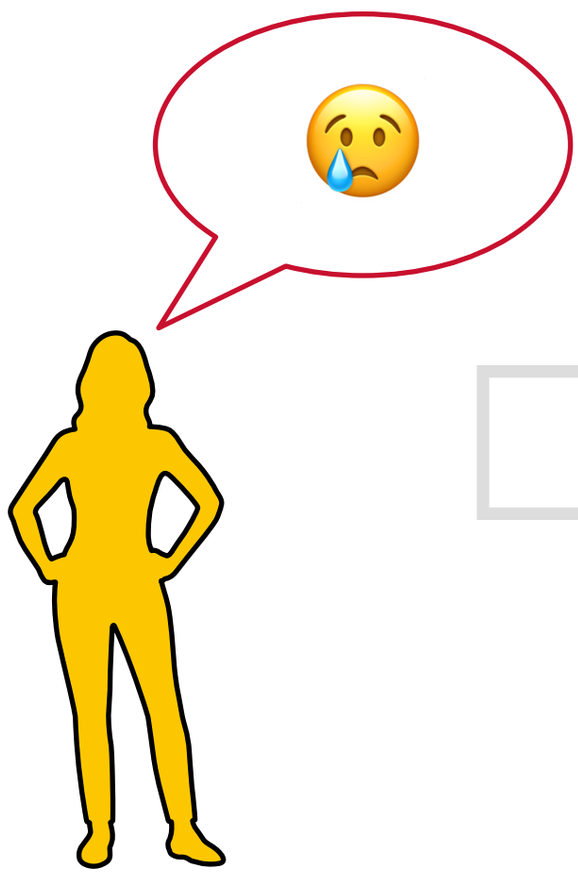
$$\Box \forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Can't use R2U2 …or can she?

Sam's Specification Revisited

Every task in the scheduler

"~~If 'task' is in the scheduler, 'task~~' shall execute within 3 seconds "

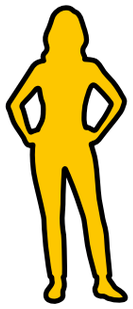


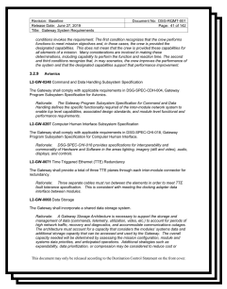
Formalize

$$\Box \forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Can't use R2U2 …or can she?

Patterns in Sam's Requirements





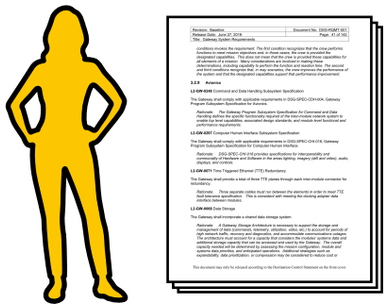
"Each request in the queue shall…"

"All active sensors shall…"

"Every task in the scheduler shall execute within 3 seconds"

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Patterns in Sam's Requirements



"Each request in the queue shall…"

"All active sensors shall…"

"Every task in the scheduler shall execute within 3 seconds"

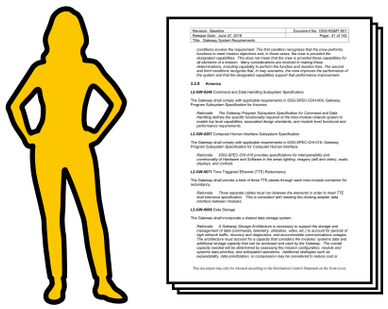
$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

- **Guarded quantifiers:**

$$\forall x : Set(x) \rightarrow \Phi$$

$$\exists x : Set(x) \wedge \Phi$$

Patterns in Sam's Requirements



"Each request in the queue shall…"

"All active sensors shall…"

"Every task in the scheduler shall execute within 3 seconds"

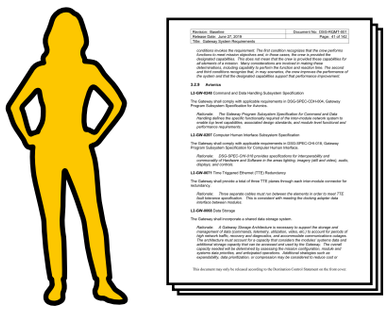
$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

- **Guarded quantifiers:**

$$\forall x : Set(x) \rightarrow \Phi$$
$$\exists x : Set(x) \wedge \Phi$$

- **Bounded Dynamic Set**: A set that changes over time and has a bounded cardinality

Patterns in Sam's Requirements



"Each request in the **queue** shall…"

"All **active sensors** shall…"

"**Every task in the scheduler shall execute within 3 seconds**"

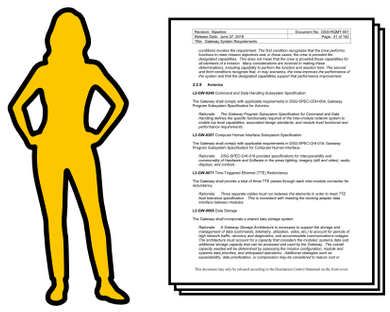
$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

- **Guarded quantifiers:**

$$\forall x : Set(x) \rightarrow \Phi$$
$$\exists x : Set(x) \wedge \Phi$$

- **Bounded Dynamic Set**: A set that changes over time and has a bounded cardinality

Patterns in Sam's Requirements



"Each request in the **queue** shall…"

"All **active sensors** shall…"

"**Every task in the scheduler shall execute within 3 seconds**"

Require

$|Sched| \leq max(Sched)$

$$\square \; \forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

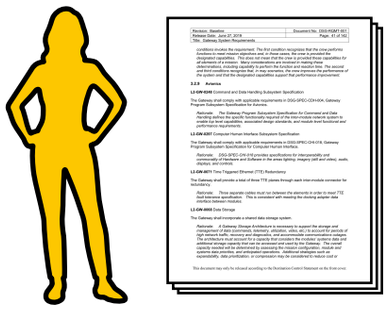
- **Guarded quantifiers:**

$$\forall x : Set(x) \rightarrow \Phi$$

$$\exists x : Set(x) \wedge \Phi$$

- **Bounded Dynamic Set**: A set that changes over time and has a bounded cardinality

Patterns in Sam's Requirements



"Each request in the **queue** shall…"

"All **active sensors** shall…"

"Every task in the scheduler shall execute within 3 seconds"

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

- **Guarded quantifiers:** $\forall x : Set(x) \rightarrow \Phi$ $\quad \exists x : Set(x) \wedge \Phi$
- **Bounded Dynamic Set**: A set that changes over time and has a bounded cardinality

$= \overline{\text{FO-MLTL}}$

FO-MLTL Quantifier Elimination

$$\cancel{\forall} t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$\bigwedge_{i \in [0,\ max(Sched)-1]} (Sched(t_i) \rightarrow \Diamond_{[0,3]} Exec(t_i))$$

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$\bigwedge_{i \in [0,\ max(Sched)-1]} (Sched(t_i) \rightarrow \Diamond_{[0,3]} Exec(t_i))$$

$$\wedge\ |Sched| \leq max(Sched)$$

Cardinality Constraint

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$\bigwedge_{i \in [0,\ max(Sched)-1]} (Sched(t_i) \rightarrow \Diamond_{[0,3]} Exec(t_i))$$

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$$
$$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

$$max(Sched) = 2$$

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$$
$$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

Need to assign meaning

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$$
$$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

Sched: 

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

Arbitrarily assign

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$$

$$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

Sched: 

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$

$$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

Sched: 

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$

$$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

Sched: A

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$

$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$$

Sched: 

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$

$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$$

Always False

Sched: 

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$

$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$$

Vacuously True

Sched: 

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$

$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$$

Vacuously True

Sched: 

Changes over time

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$$

$$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

	0	1	2	3	4
Sched					

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$$

$$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

	0	1	2	3	4
Sched	■■	■■	■■	■■	■■

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$$

$$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

	0	1	2	3	4
Sched	A	A B		C	D E

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$
$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$$

	0	1	2	3	4
Sched	A	A B		C	D E

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$
$$(Sched(B) \rightarrow \Diamond_{[0,3]} Exec(B))$$

	0	1	2	3	4
Sched	A	A B		C	D E

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp)) \wedge$$
$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$$

	0	1	2	3	4
Sched	A	A B		C	D E

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$$
$$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

Instantiate

$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$
$(Sched(\bot) \rightarrow \Diamond_{[0,3]} Exec(\bot))$

$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$
$(Sched(B) \rightarrow \Diamond_{[0,3]} Exec(B))$

$(Sched(\bot) \rightarrow \Diamond_{[0,3]} Exec(\bot)) \wedge$
$(Sched(\bot) \rightarrow \Diamond_{[0,3]} Exec(\bot))$

...

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$$

$$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

Instantiate

$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$
$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$

$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$
$(Sched(B) \rightarrow \Diamond_{[0,3]} Exec(B))$

$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp)) \wedge$
$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$

...

MLTL Monitors (.bin **)**

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$$

$$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

Instantiate

How many?

$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$
$(Sched(\bot) \rightarrow \Diamond_{[0,3]} Exec(\bot))$

$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$
$(Sched(B) \rightarrow \Diamond_{[0,3]} Exec(B))$

$(Sched(\bot) \rightarrow \Diamond_{[0,3]} Exec(\bot)) \wedge$
$(Sched(\bot) \rightarrow \Diamond_{[0,3]} Exec(\bot))$

...

MLTL Monitors (.bin **)**

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll **Will know answer by time 3**

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$
$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$$

	0	1	2	3	4
Sched	A	A B		C	D E

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$

$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$
$(Sched(\bot) \rightarrow \Diamond_{[0,3]} Exec(\bot))$

$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$
$(Sched(B) \rightarrow \Diamond_{[0,3]} Exec(B))$

$(Sched(\bot) \rightarrow \Diamond_{[0,3]} Exec(\bot)) \wedge$
$(Sched(\bot) \rightarrow \Diamond_{[0,3]} Exec(\bot))$

$(Sched(\bot) \rightarrow \Diamond_{[0,3]} Exec(\bot)) \wedge$
$(Sched(C) \rightarrow \Diamond_{[0,3]} Exec(C))$

	0	1	2	3	4
Sched	A	A B		C	D E

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$

$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$$

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$

$$(Sched(B) \rightarrow \Diamond_{[0,3]} Exec(B))$$

$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp)) \wedge$$

$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$$

$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp)) \wedge$$

$$(Sched(C) \rightarrow \Diamond_{[0,3]} Exec(C))$$

	0	1	2	3	4
Sched	A	A B		C	D E

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

Re-use this space!

$$[Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$$

$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$
$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$

$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge$
$(Sched(B) \rightarrow \Diamond_{[0,3]} Exec(B))$

$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp)) \wedge$
$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$

$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp)) \wedge$
$(Sched(C) \rightarrow \Diamond_{[0,3]} Exec(C))$

	0	1	2	3	4
Sched	A	A B		C	D E

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

Unroll

Re-use this space!

Only need space for 4 formulas

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge (Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$$

$$(Sched(A) \rightarrow \Diamond_{[0,3]} Exec(A)) \wedge (Sched(B) \rightarrow \Diamond_{[0,3]} Exec(B))$$

$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp)) \wedge (Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp))$$

$$(Sched(\perp) \rightarrow \Diamond_{[0,3]} Exec(\perp)) \wedge (Sched(C) \rightarrow \Diamond_{[0,3]} Exec(C))$$

$$(Sched(D) \rightarrow \Diamond_{[0,3]} Exec(D)) \wedge (Sched(E) \rightarrow \Diamond_{[0,3]} Exec(E))$$

	0	1	2	3	4
Sched	A	A B		C	D E

FO-MLTL **Quantifier Elimination**

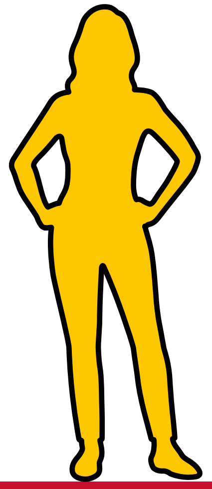
$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \land (Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \land (Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \land (Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$

$$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \land (Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$$



FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

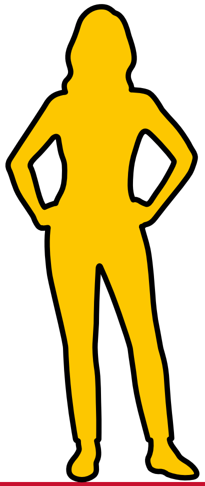
$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$
$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$

$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$
$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$

$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$
$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$

$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$
$(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$

(Quantifier-free) MLTL formulas

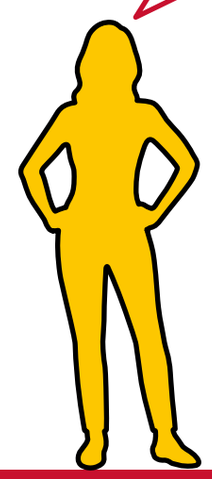


FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$ $(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$	$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$ $(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$	$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$ $(Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$	$(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge$ $(Sched(t_1)$

Yay!



(Quantifier-free) MLTL formulas

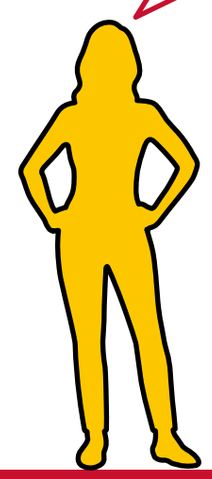
Can monitor using R2U2!

FO-MLTL Quantifier Elimination

$$\forall t : Sched(t) \rightarrow \Diamond_{[0,3]} Exec(t)$$

- $(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge (Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$
- $(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge (Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$
- $(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge (Sched(t_1) \rightarrow \Diamond_{[0,3]} Exec(t_1))$
- $(Sched(t_0) \rightarrow \Diamond_{[0,3]} Exec(t_0)) \wedge (Sched(t_1)$

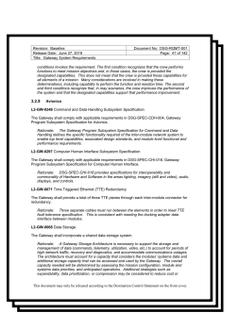
Yay!



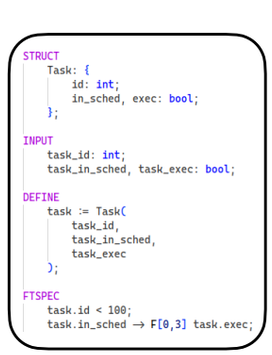
(Quantifier-free) MLTL formulas

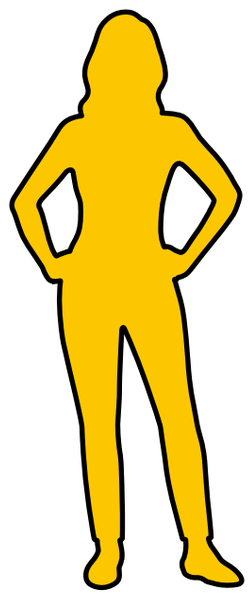
Can monitor using R2U2!***

R2U2 + C2PO Approach

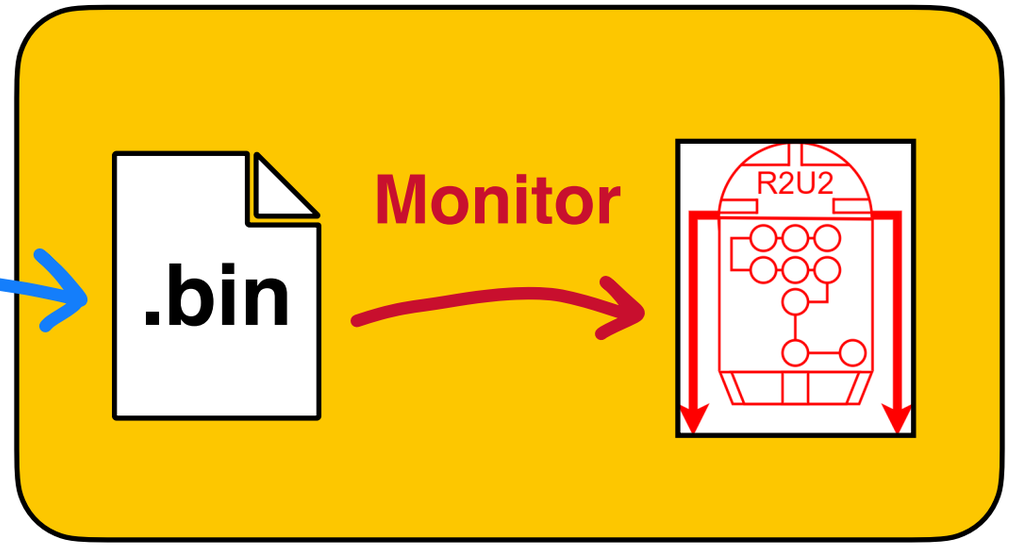


Formaliz

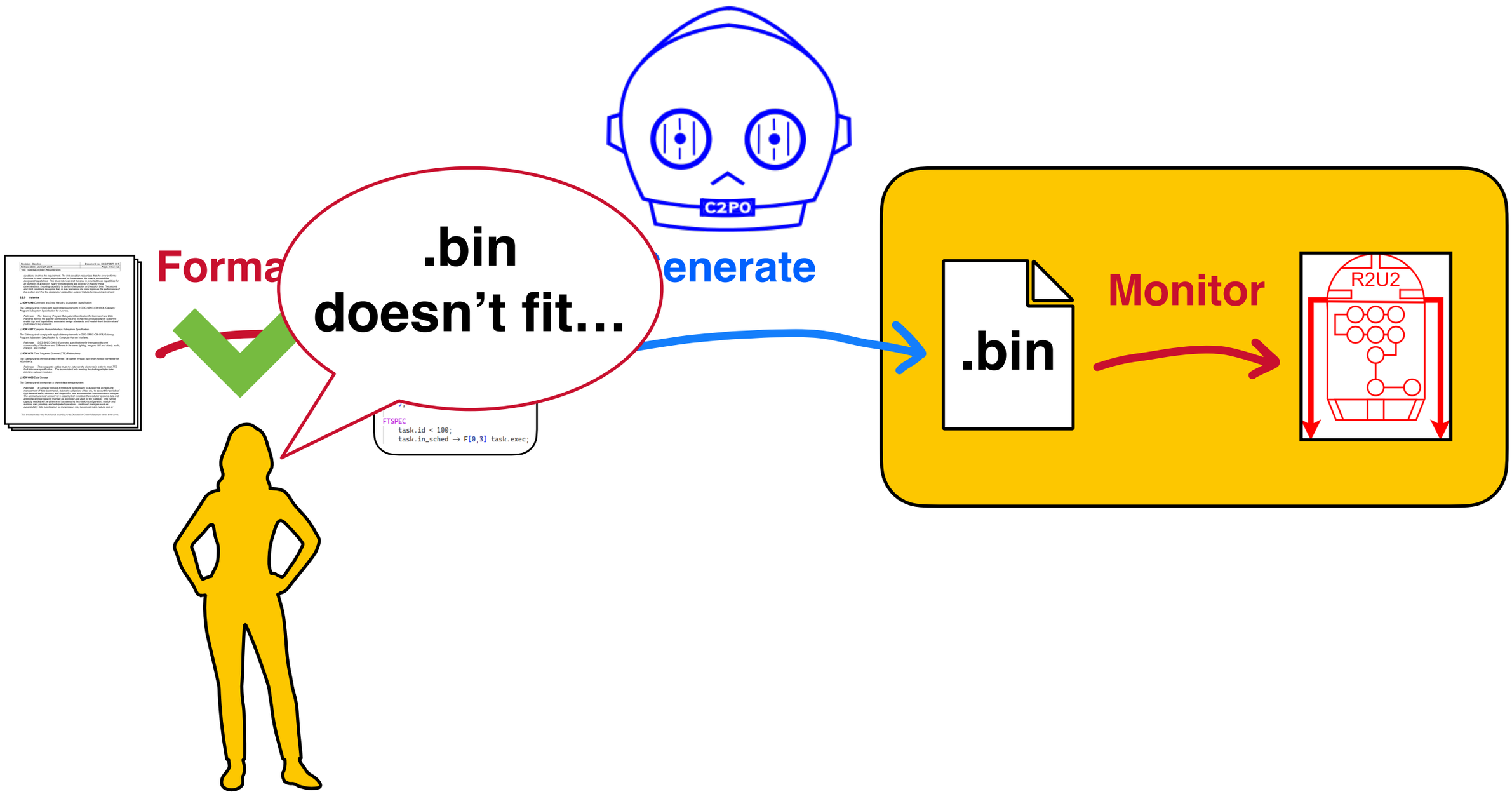




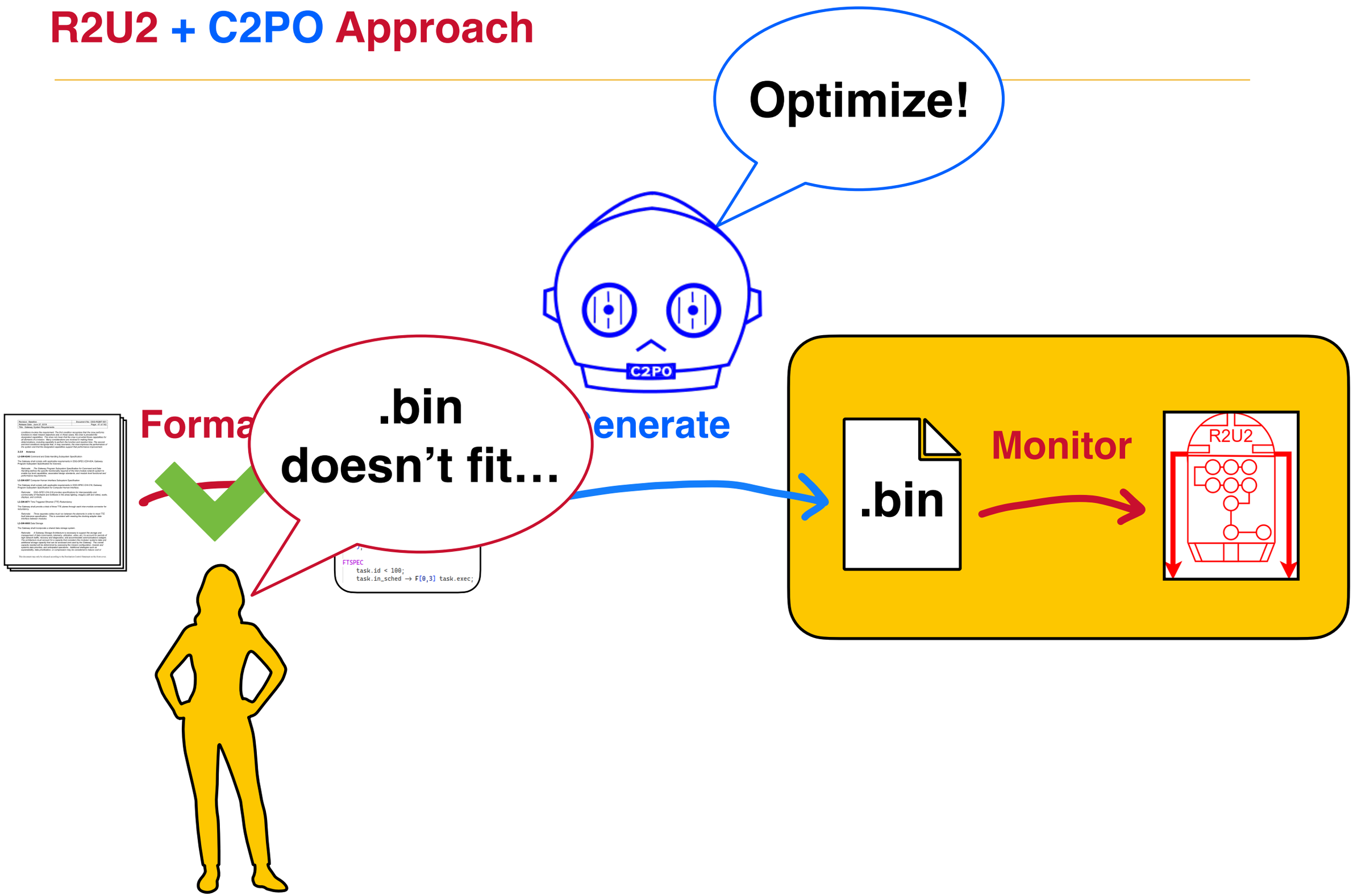
Generate



R2U2 + C2PO Approach

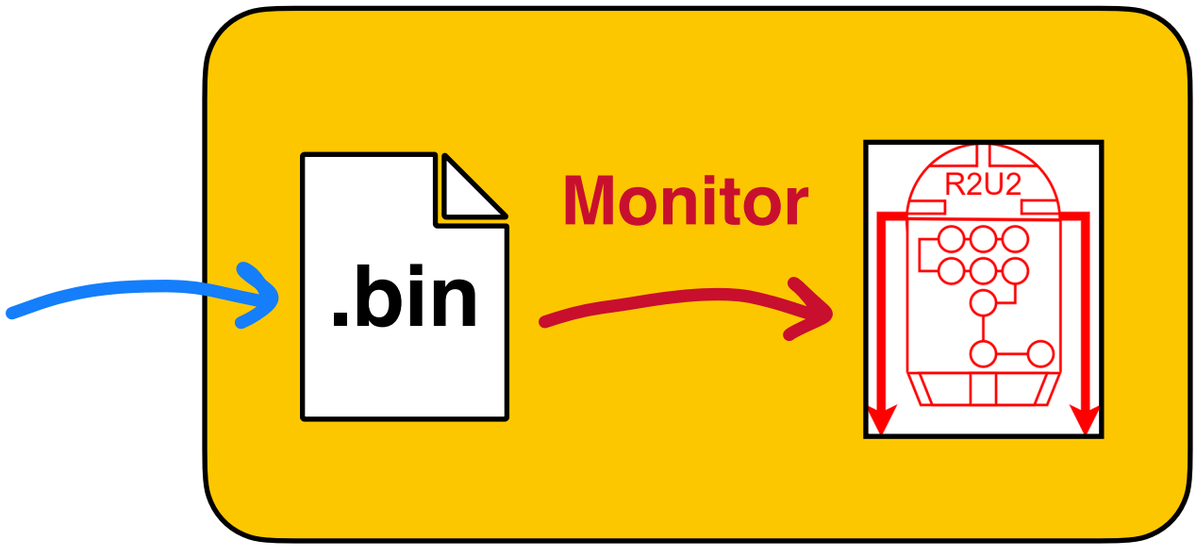


R2U2 + C2PO Approach



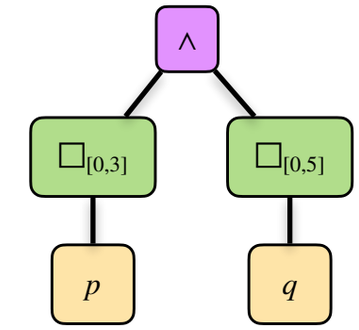
How does R2U2 Encode MLTL? [18]

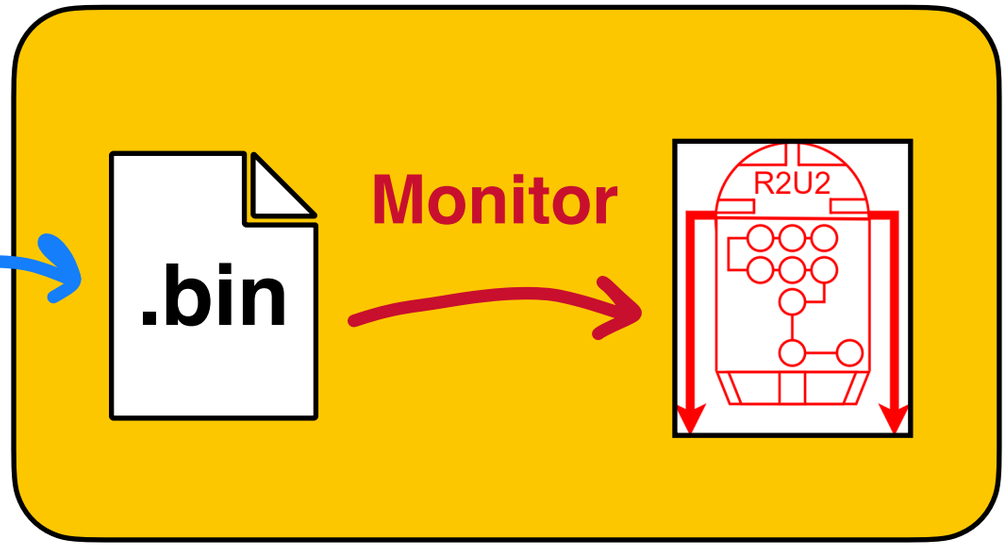
$$\square_{[0,3]}\, p \wedge \square_{[0,5]} q$$



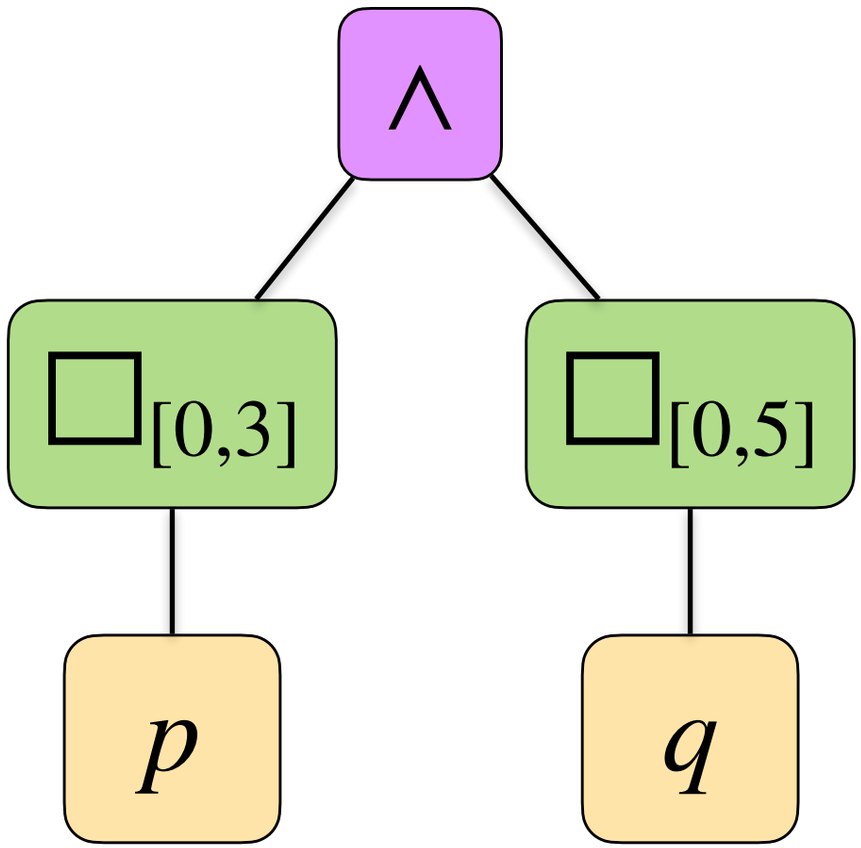
How does R2U2 Encode MLTL? [18]

$\square_{[0,3]}\, p \wedge \square_{[0,5]} q$

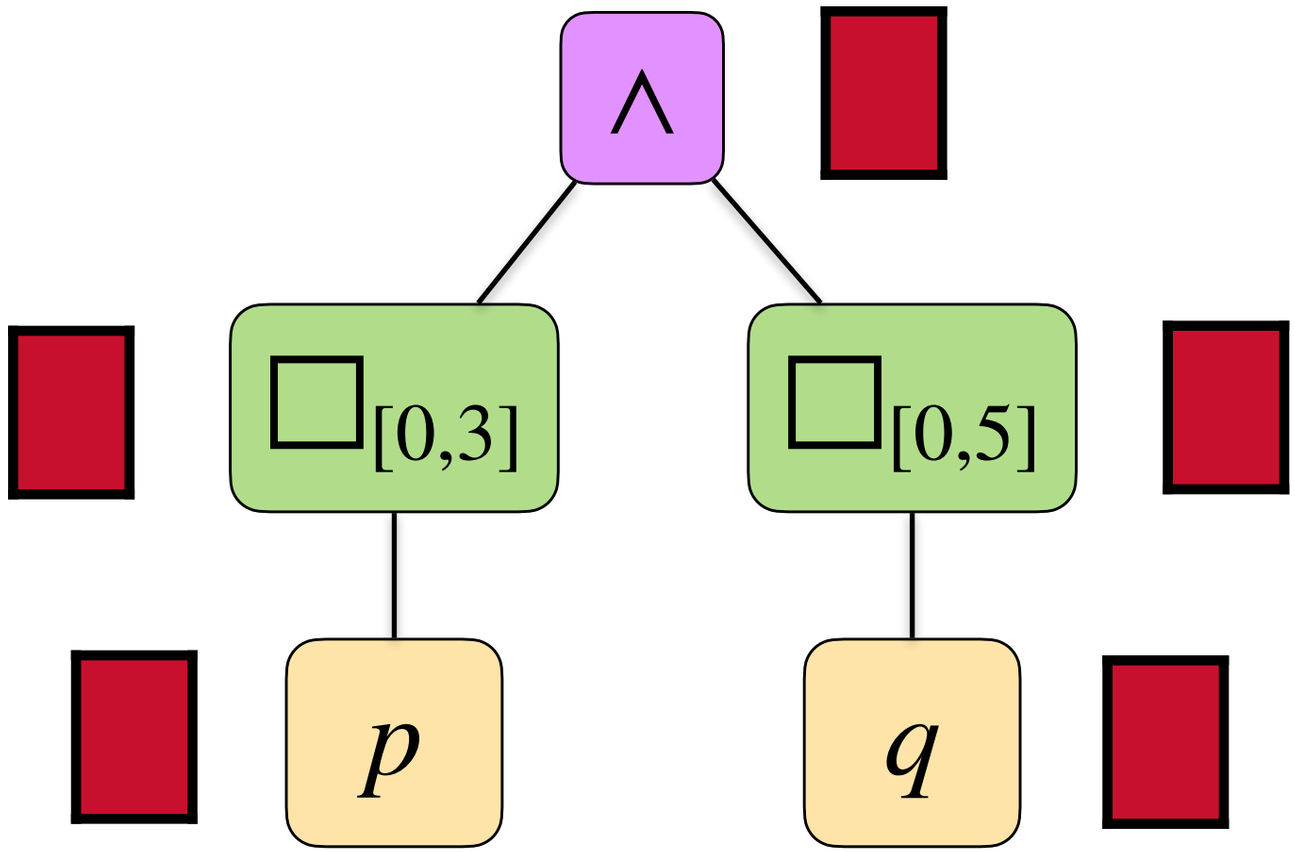




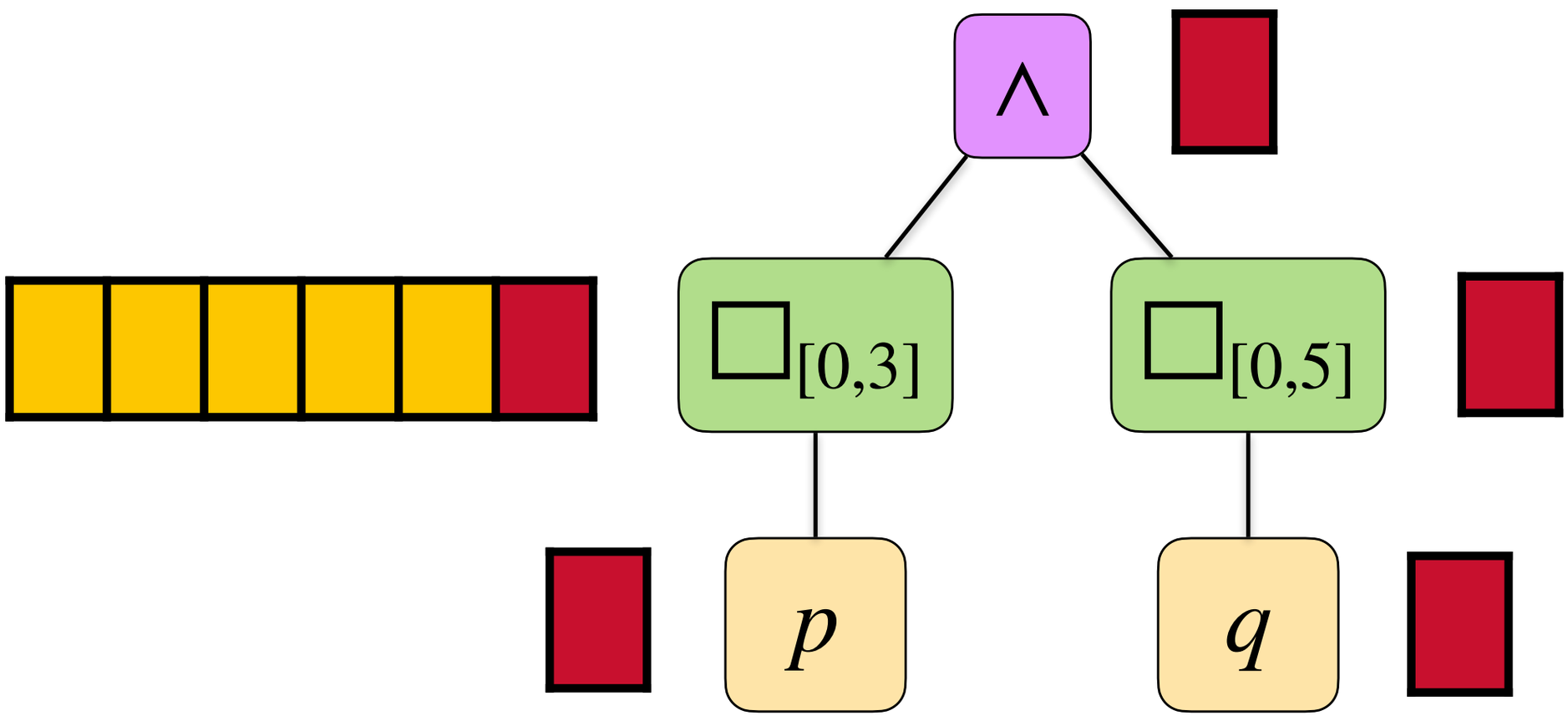
How does R2U2 Encode MLTL? [18]



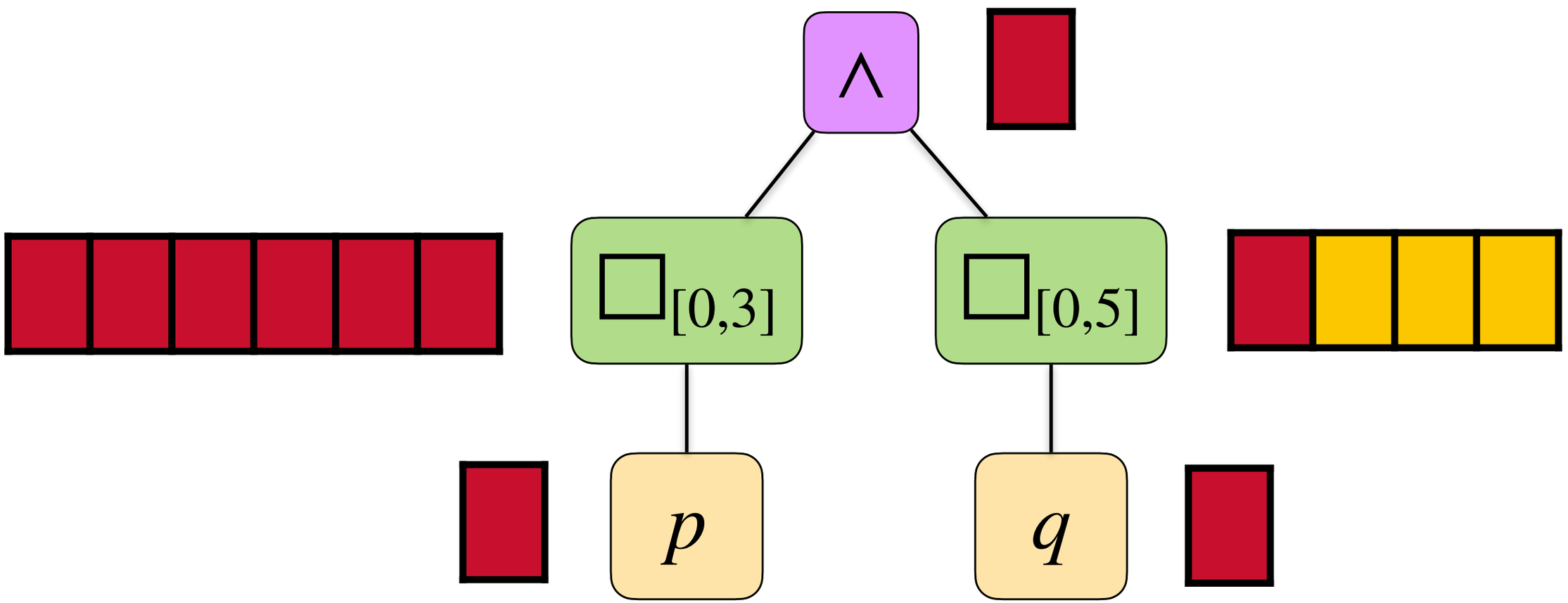
How does R2U2 Encode MLTL? [18]



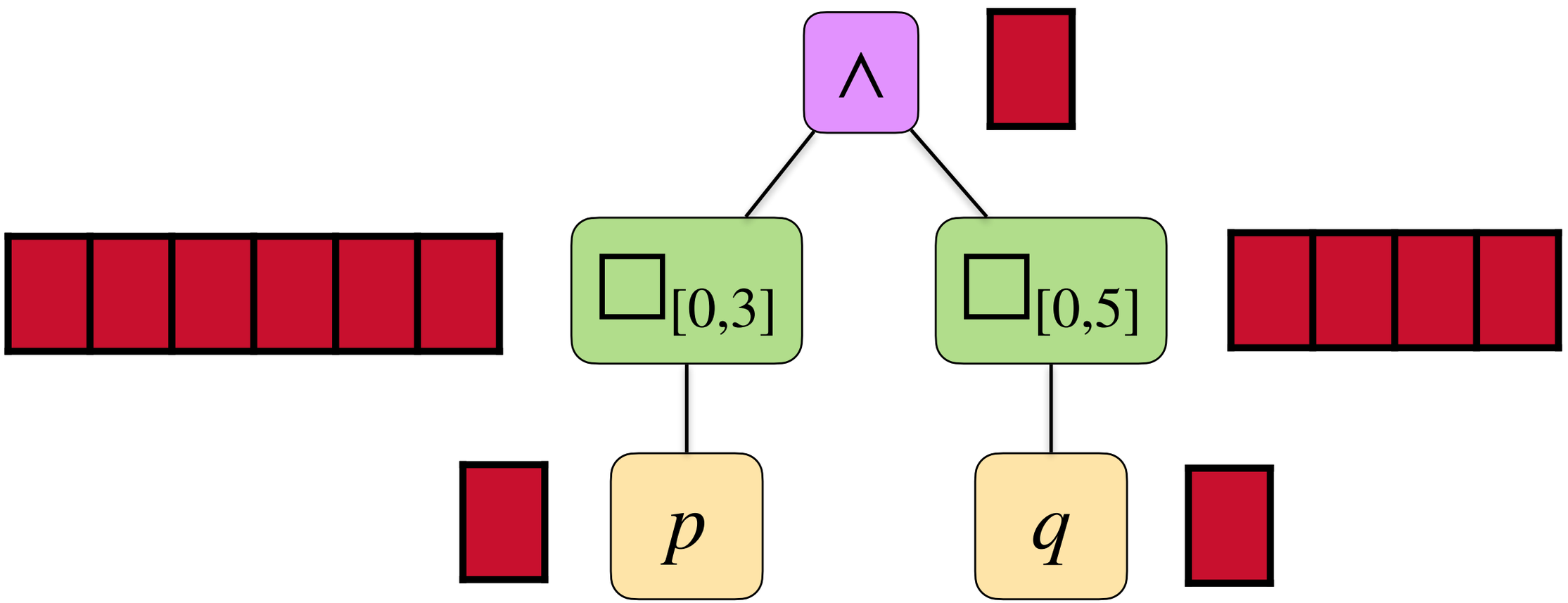
How does R2U2 Encode MLTL? [18]



How does R2U2 Encode MLTL? [18]



How does R2U2 Encode MLTL? [18]



| buffer | ∼ upper bounds of siblings

- **Reduce number of nodes**
- **Reduce sum of upper bounds**

MLTL Rewrite Rules

$$\square_{[0,3]} p \wedge \square_{[0,5]} q$$

MLTL Rewrite Rules

$$\square_{[0,3]} p \wedge \square_{[0,5]} q$$

Share the interval [0,3]

MLTL Rewrite Rules

$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$

$$\Rightarrow$$

$$\Box_{[0,3]} (\Box_{[0,0]} p \wedge \Box_{[0,2]} q)$$

Factor out common interval

MLTL Rewrite Rules

$$\square_{[0,3]} p \wedge \square_{[0,5]} q$$

$$\Rightarrow$$

$$\square_{[0,3]} (\square_{[0,0]} p \wedge \square_{[0,2]} q)$$

$$\square_{[0,0]} p \equiv p$$

MLTL Rewrite Rules

$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$

$$\Rightarrow$$

$$\Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

MLTL Rewrite Rules

$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$

	0	1	2	3	4	5
p						
q						

$$=$$

$$\Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

MLTL Rewrite Rules

$$\square_{[0,3]} p \wedge \square_{[0,5]} q$$

	0	1	2	3	4	5
p						
q						

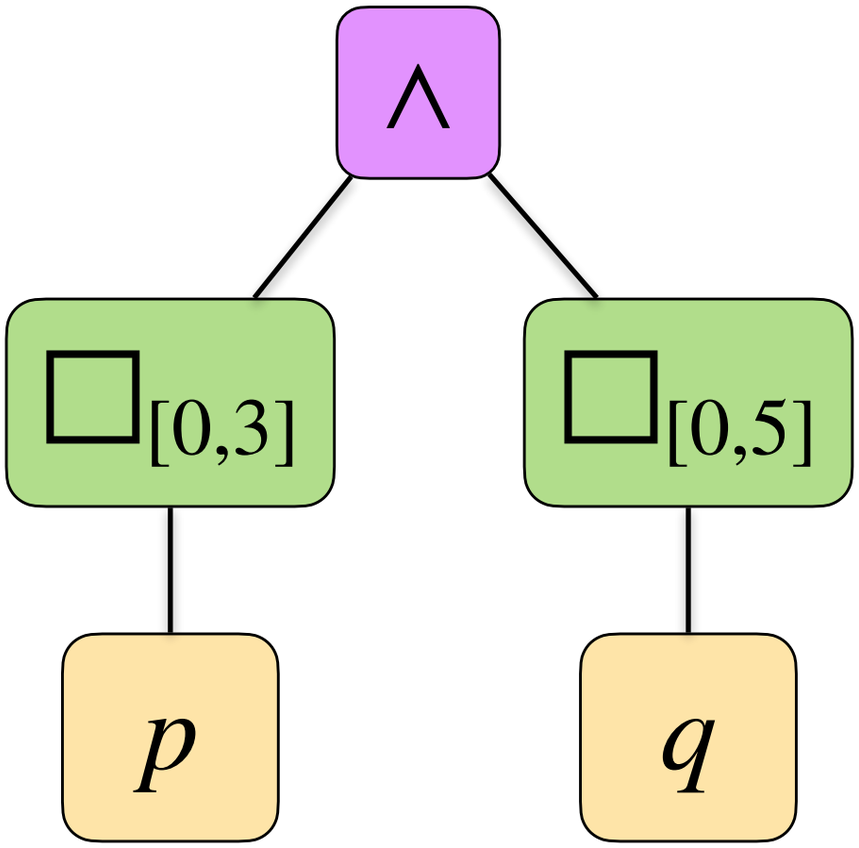
$$=$$

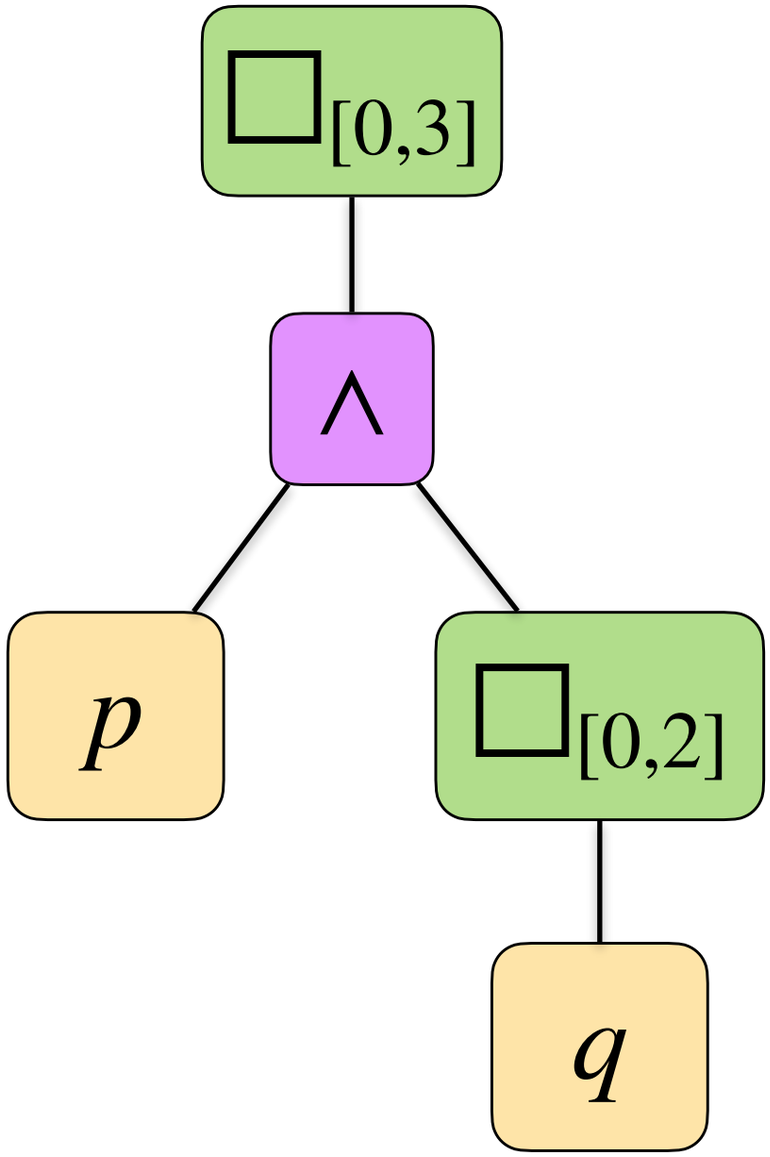
$$\square_{[0,3]} (p \wedge \square_{[0,2]} q)$$

	0	1	2	3	4	5
p						
q						

MLTL Rewrite Rules

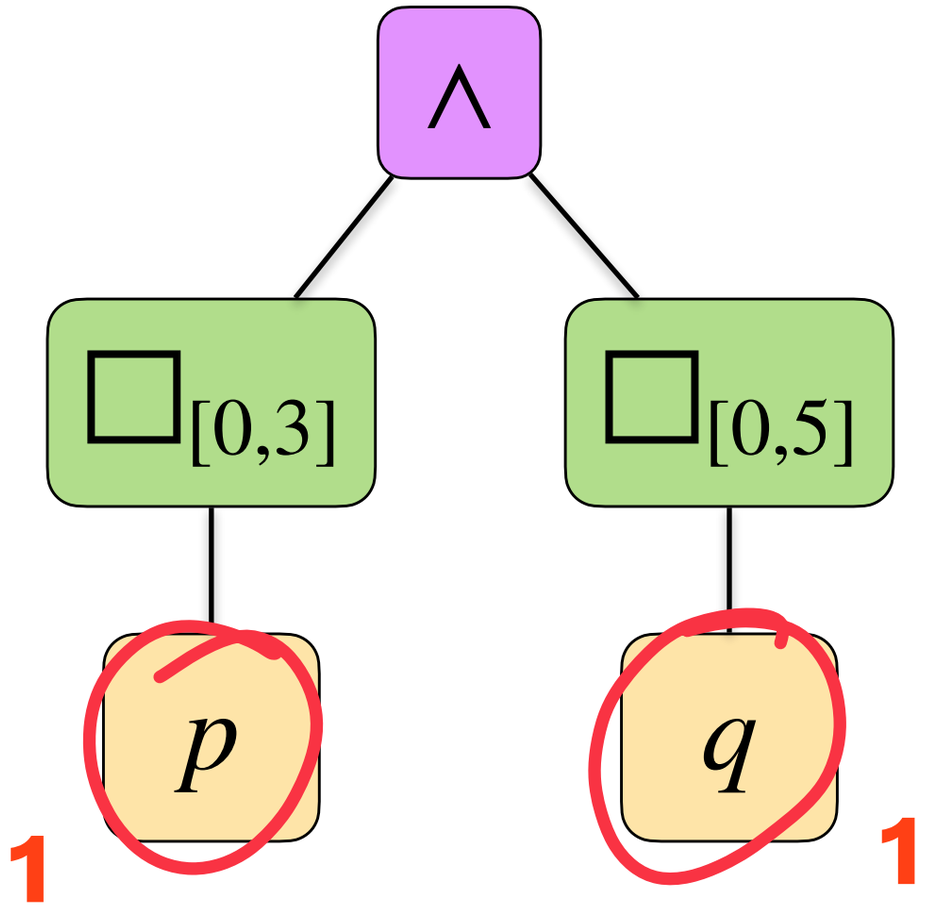
$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q \quad \Rightarrow \quad \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

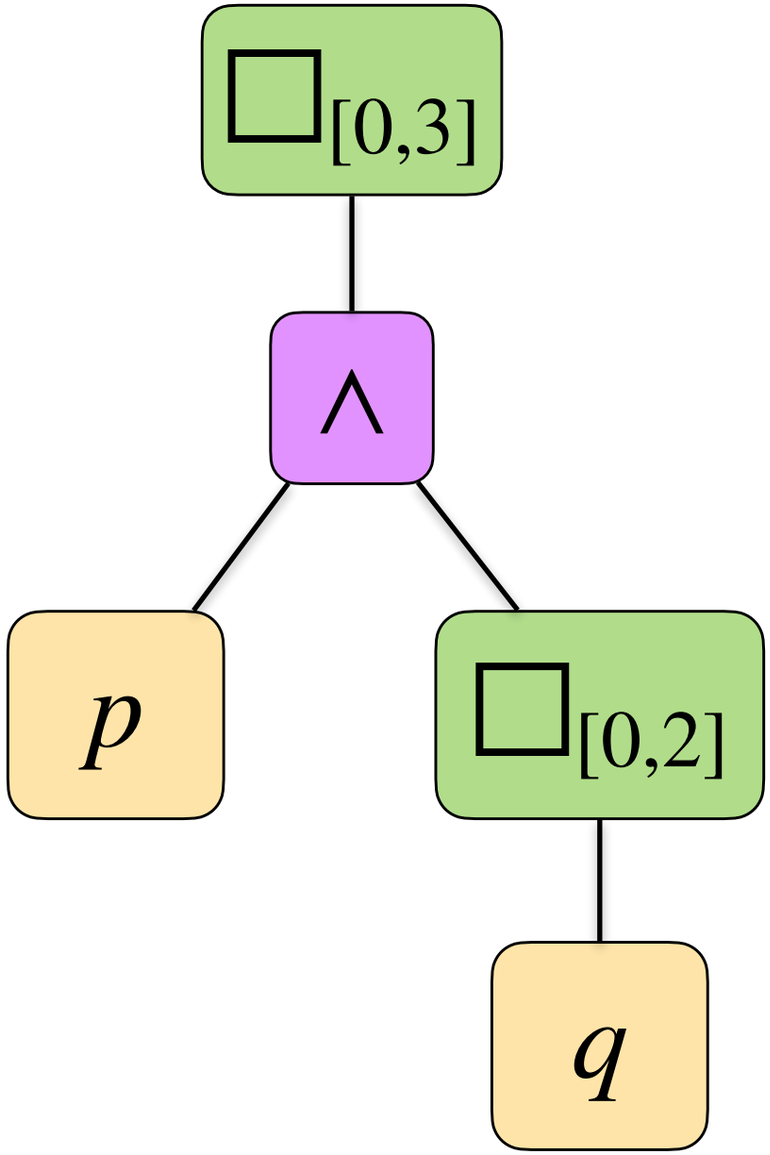




MLTL Rewrite Rules

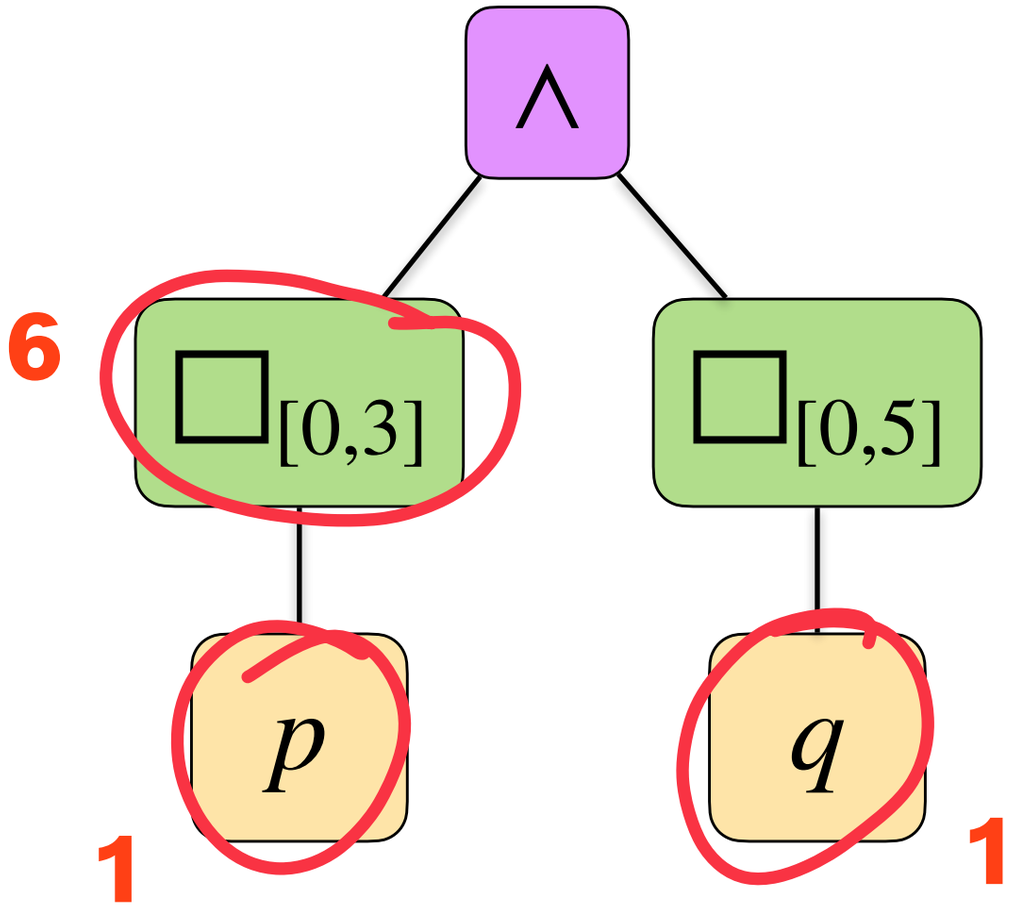
$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q \quad \Rightarrow \quad \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

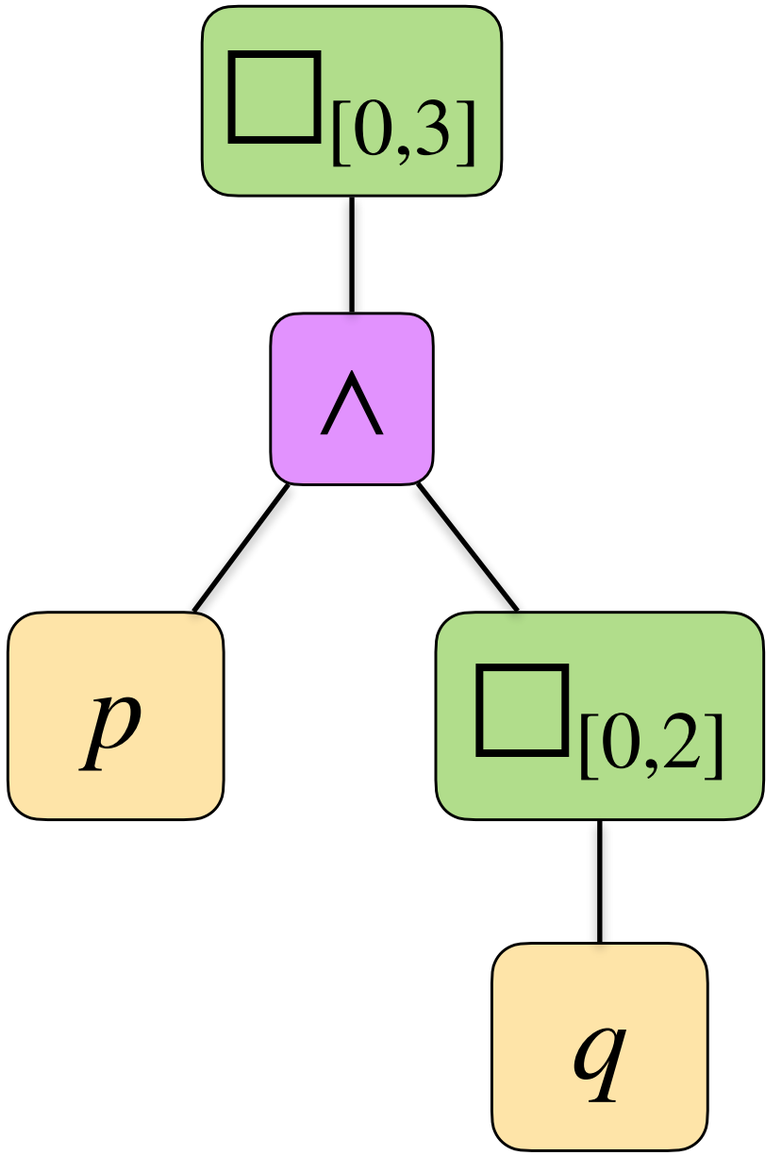




MLTL Rewrite Rules

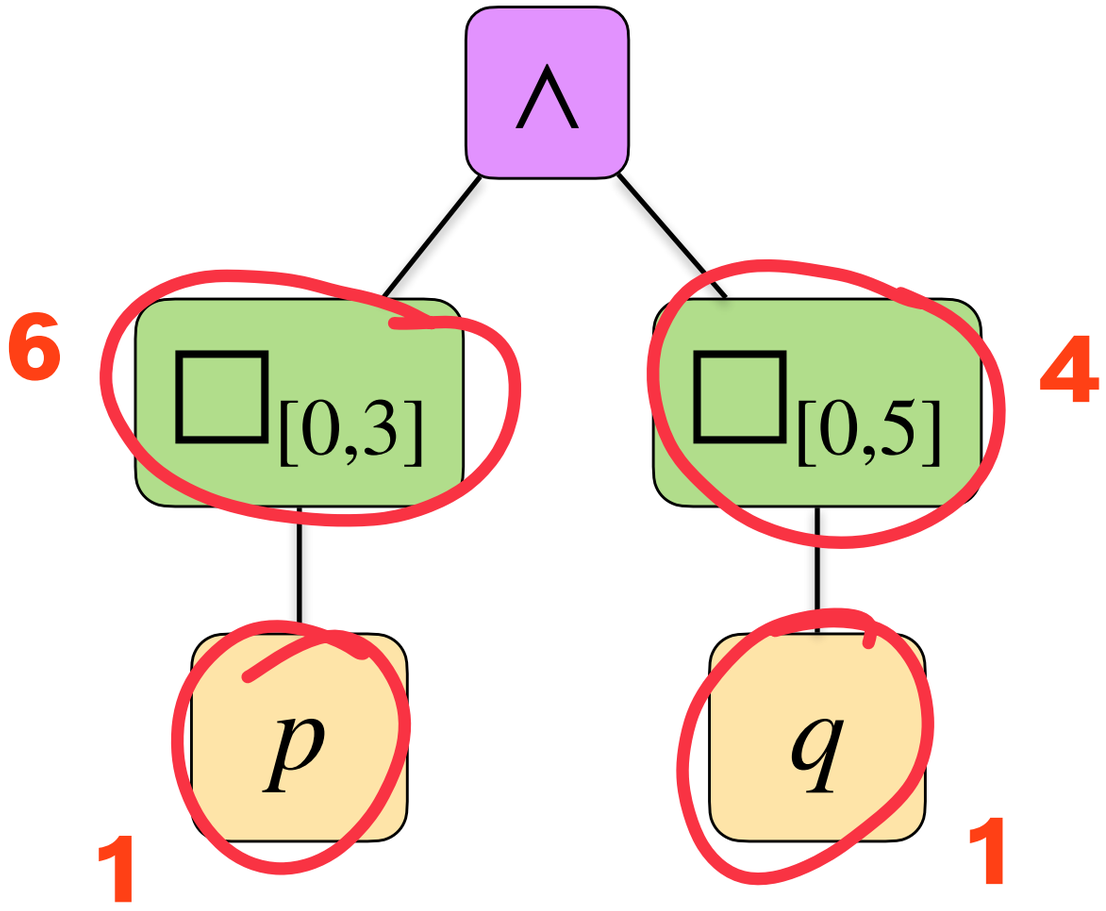
$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q \quad \Rightarrow \quad \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

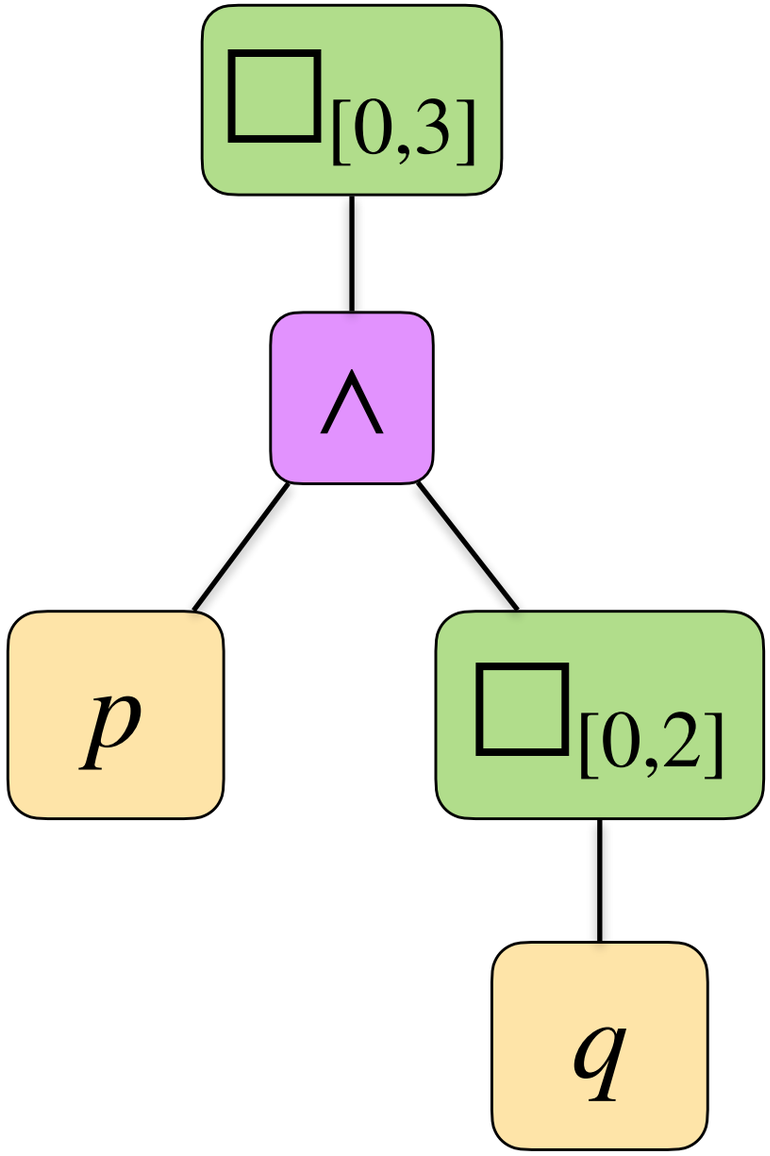




MLTL Rewrite Rules

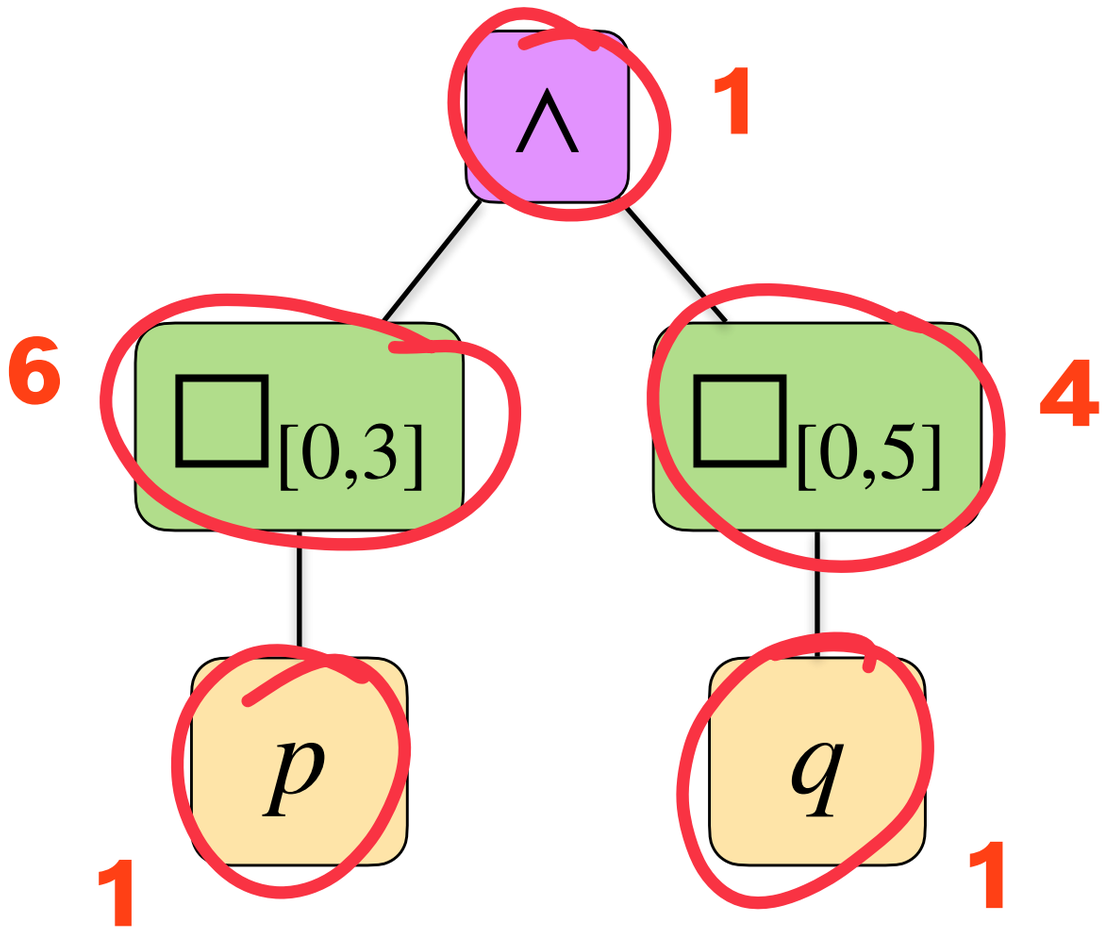
$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q \quad \Rightarrow \quad \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

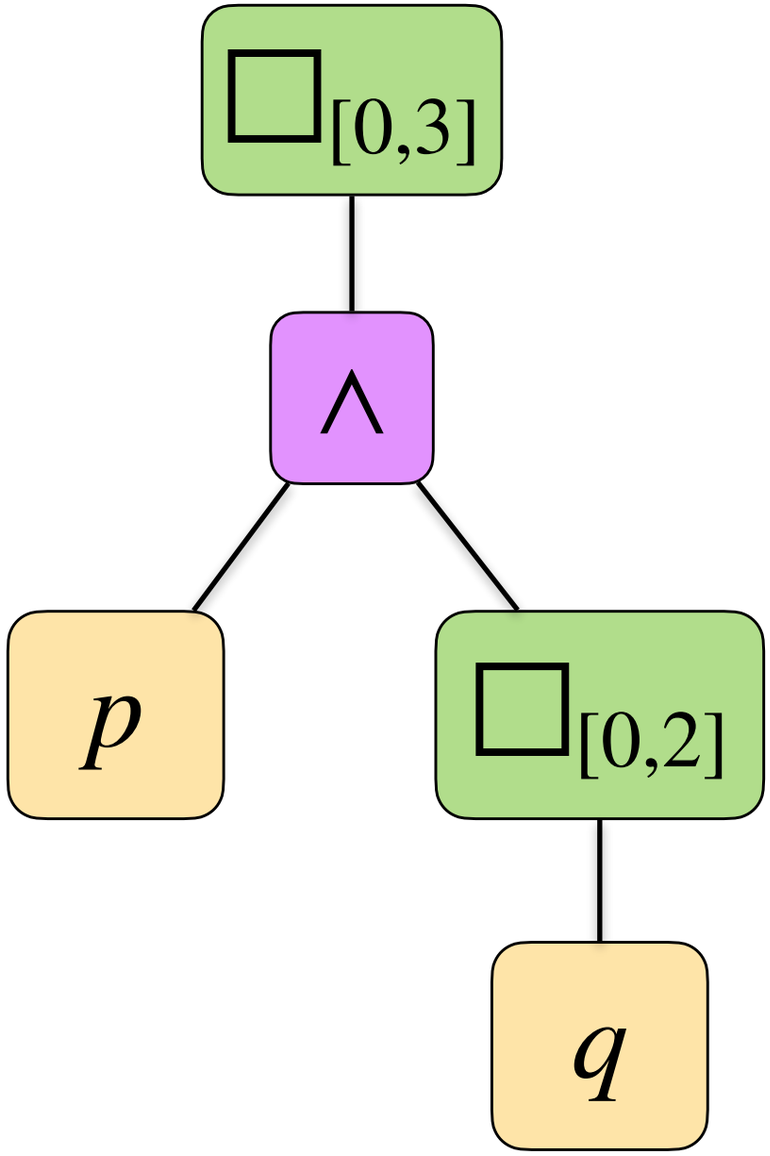




MLTL Rewrite Rules

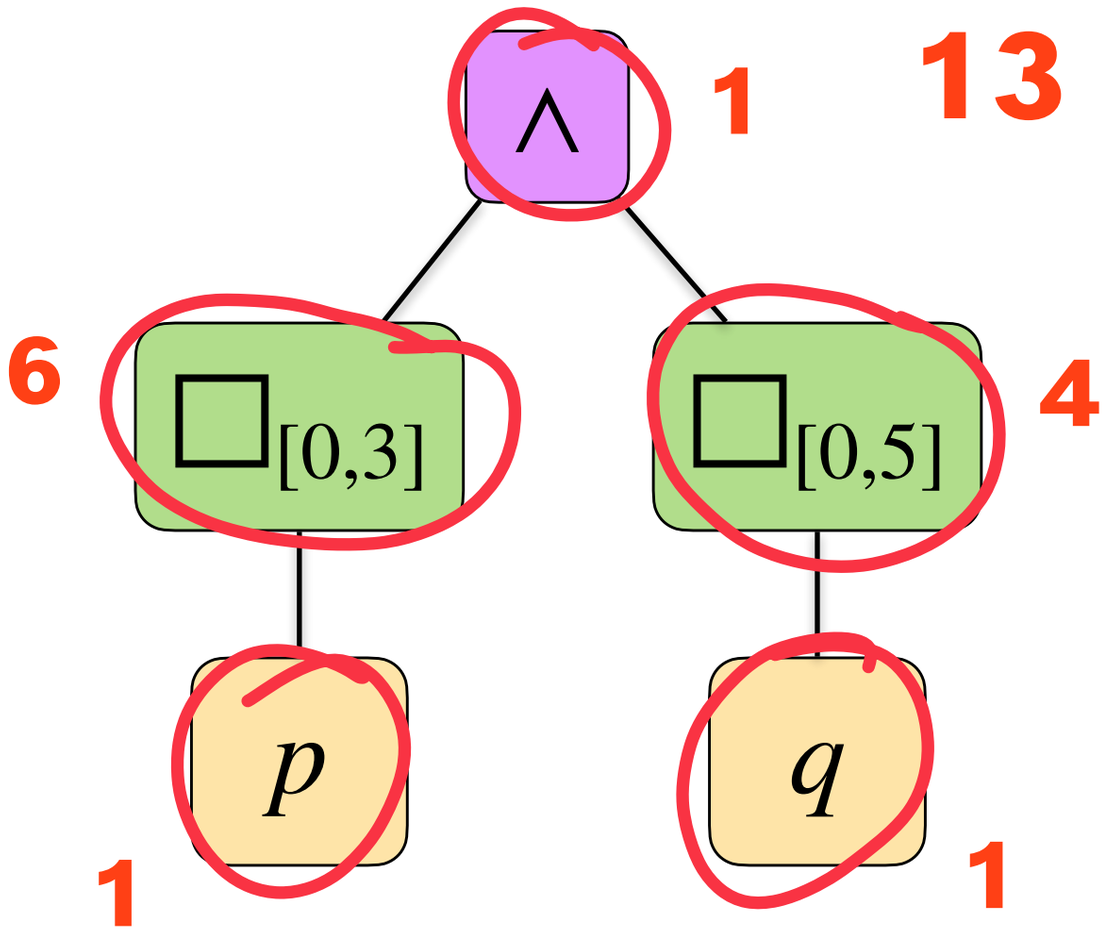
$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q \quad \Rightarrow \quad \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

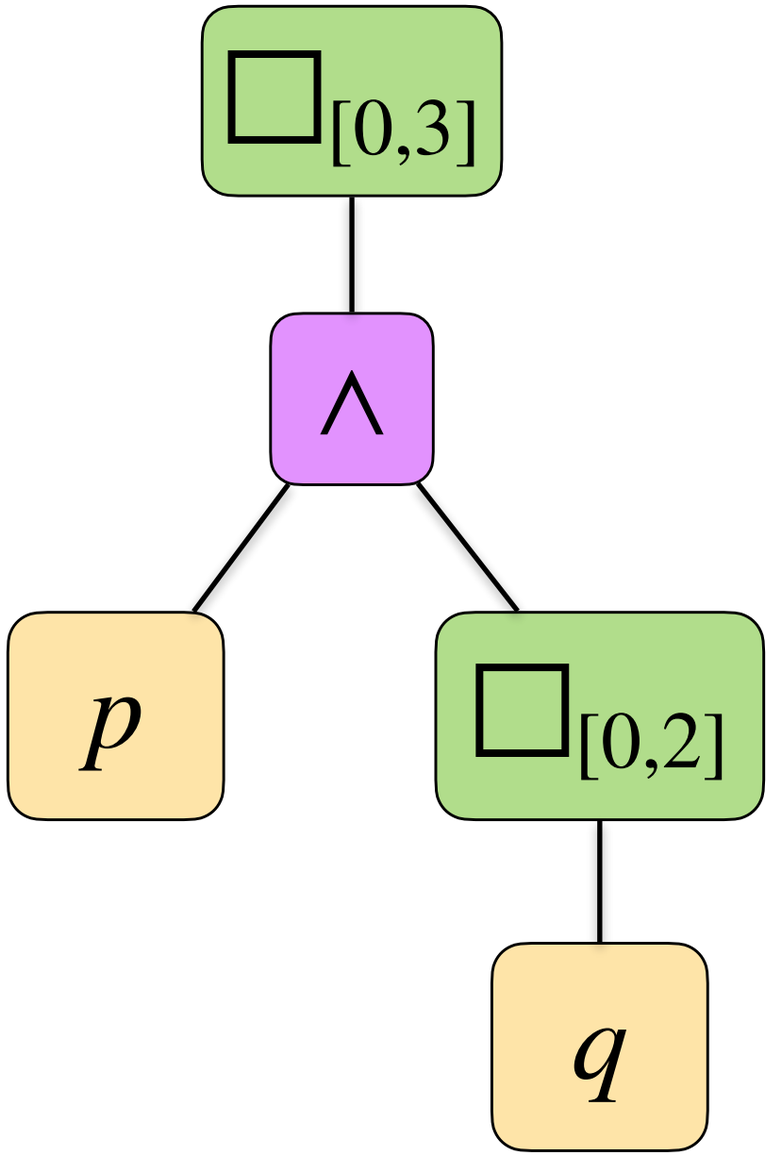




MLTL Rewrite Rules

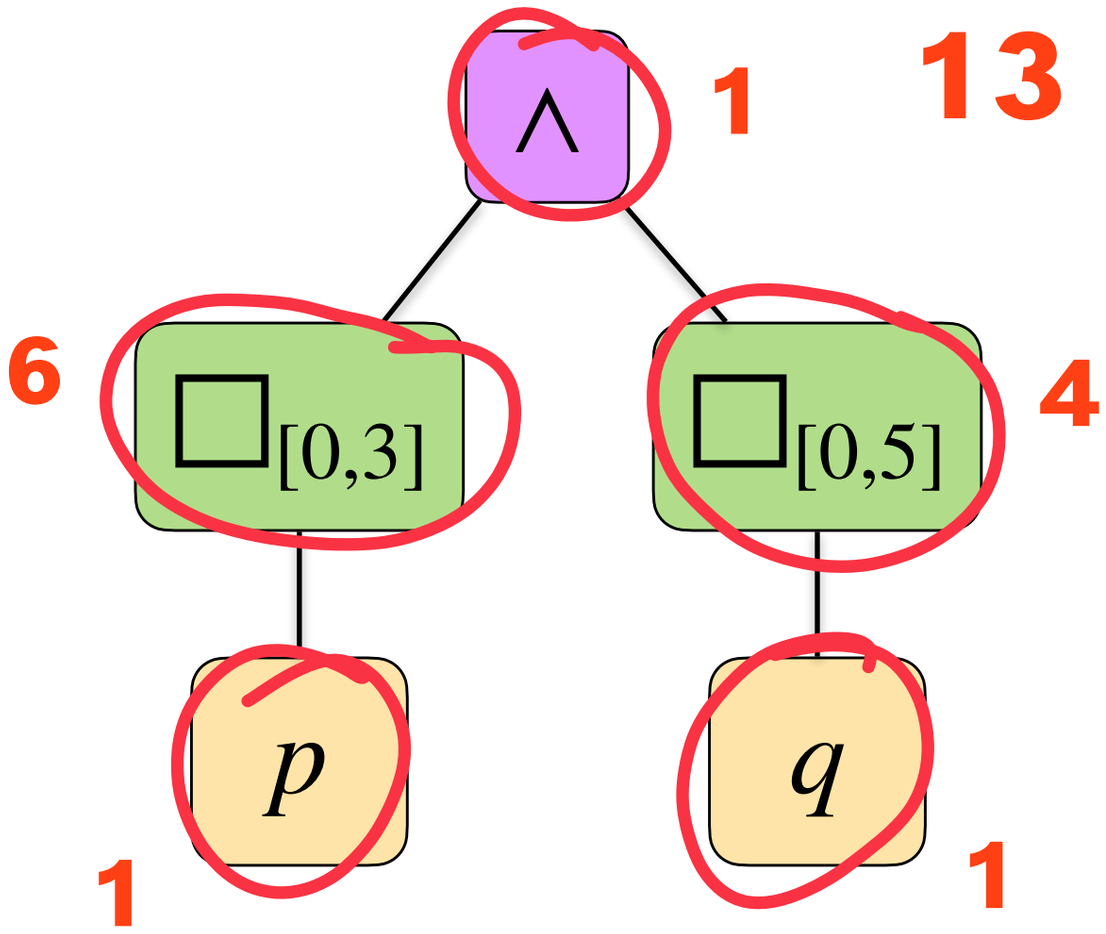
$$\square_{[0,3]} p \wedge \square_{[0,5]} q \quad \Rightarrow \quad \square_{[0,3]} (p \wedge \square_{[0,2]} q)$$

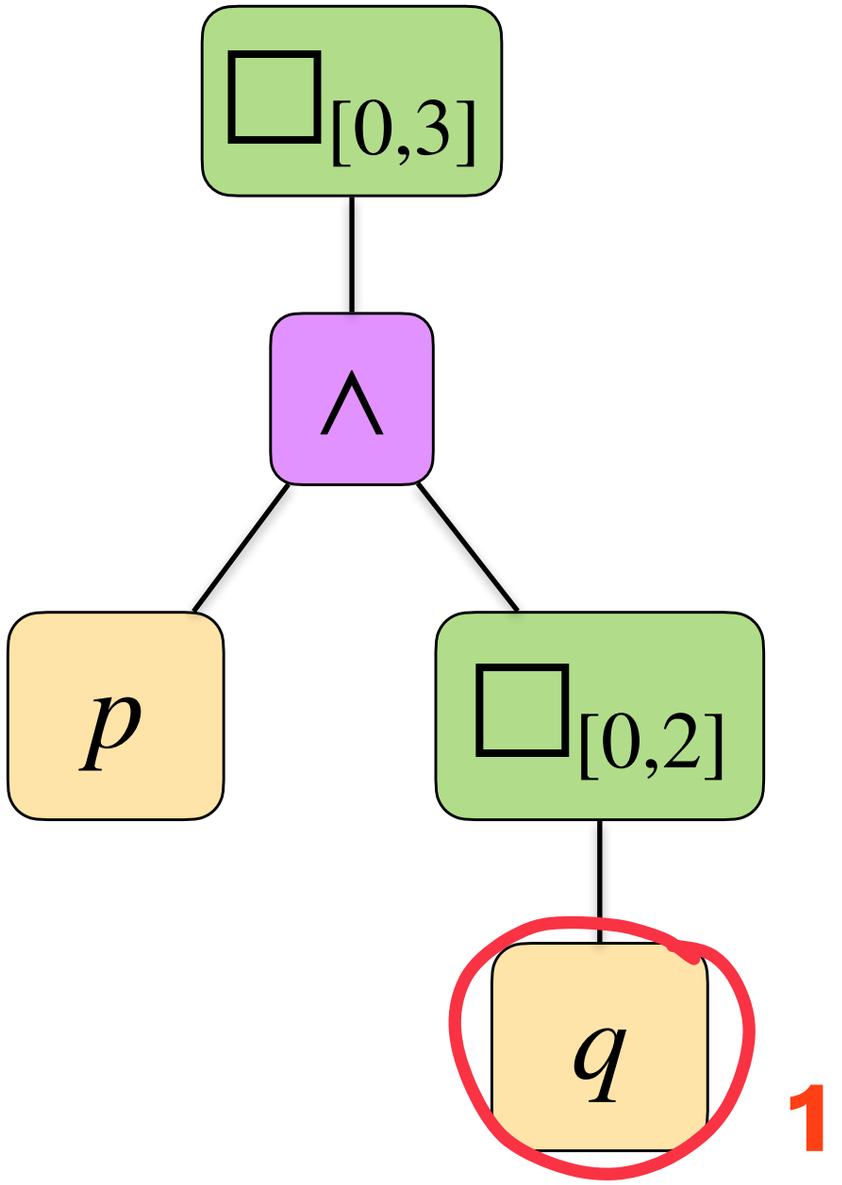




MLTL Rewrite Rules

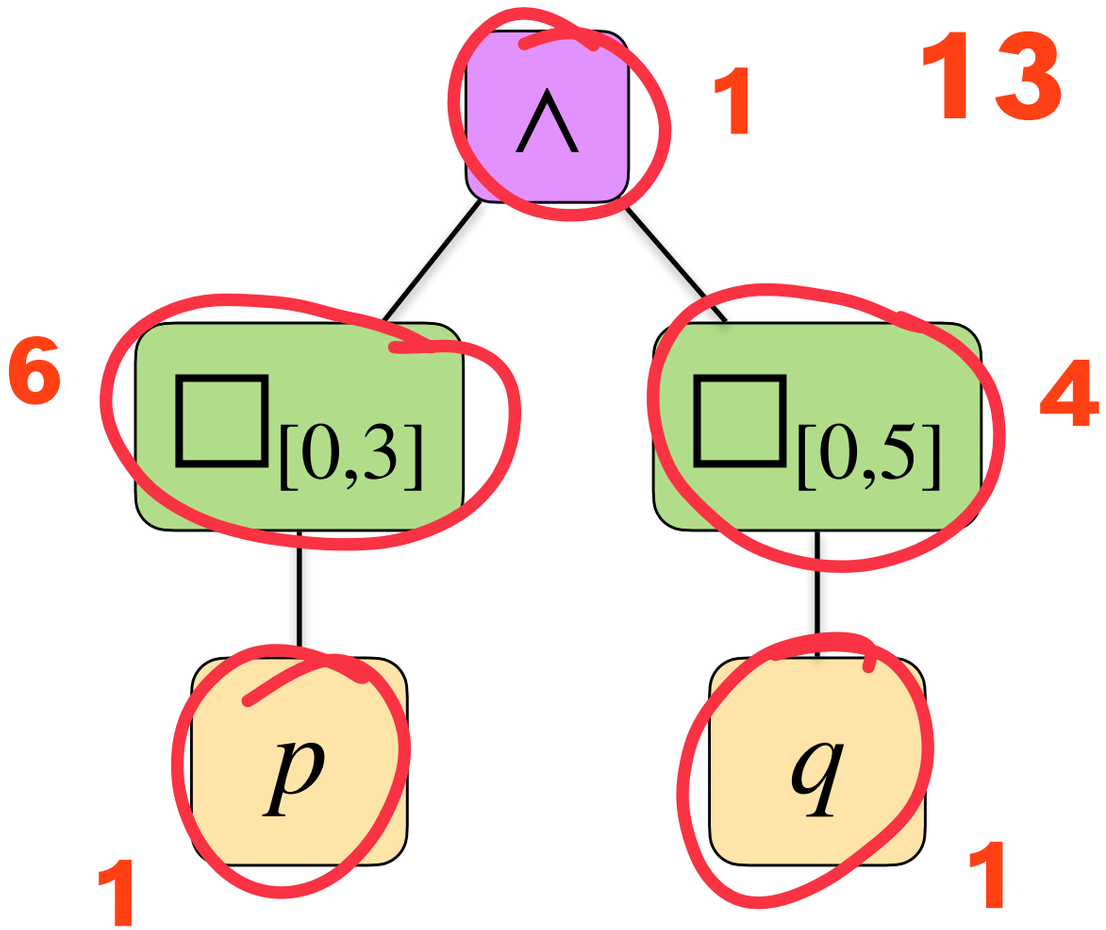
$$\square_{[0,3]} p \wedge \square_{[0,5]} q \quad \Rightarrow \quad \square_{[0,3]} (p \wedge \square_{[0,2]} q)$$

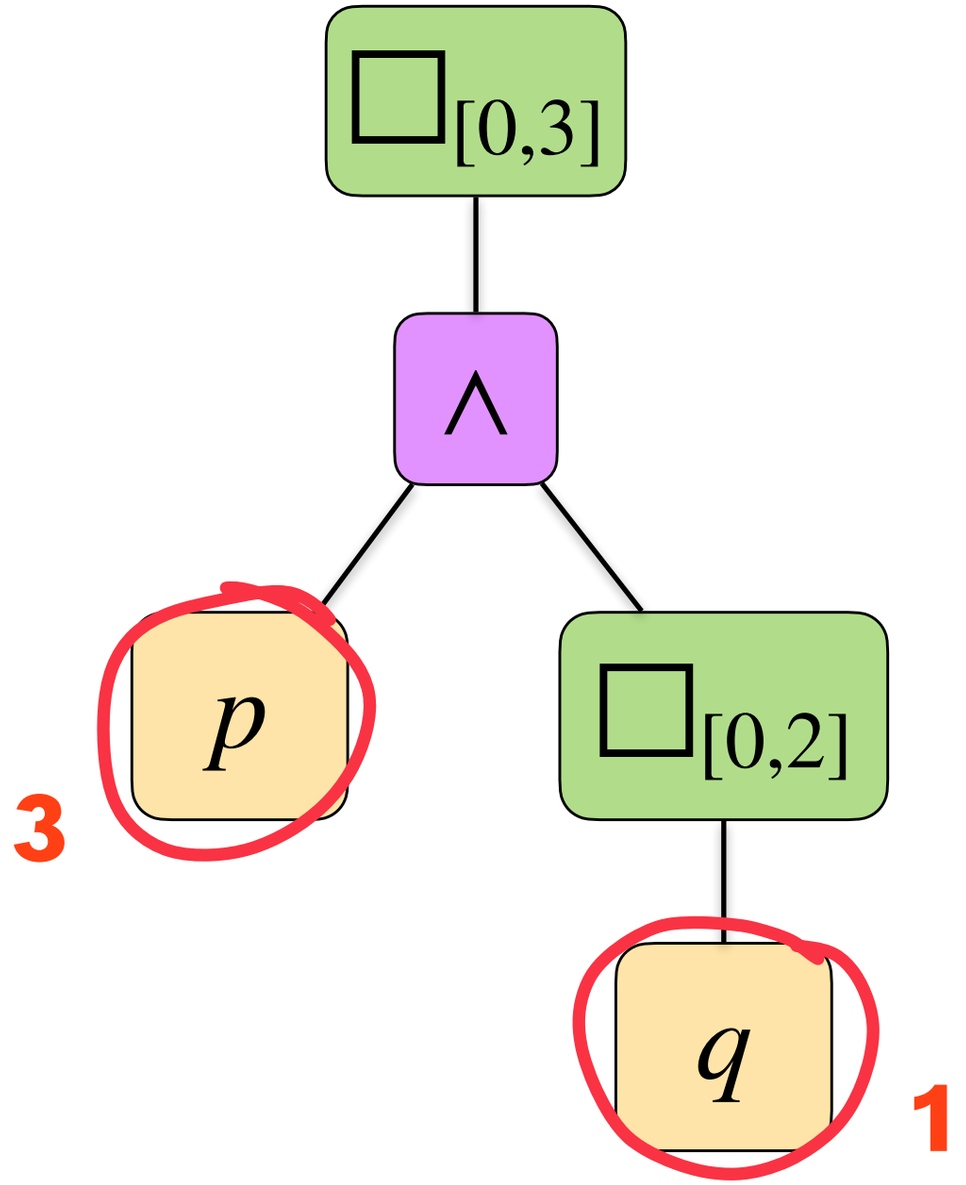




MLTL Rewrite Rules

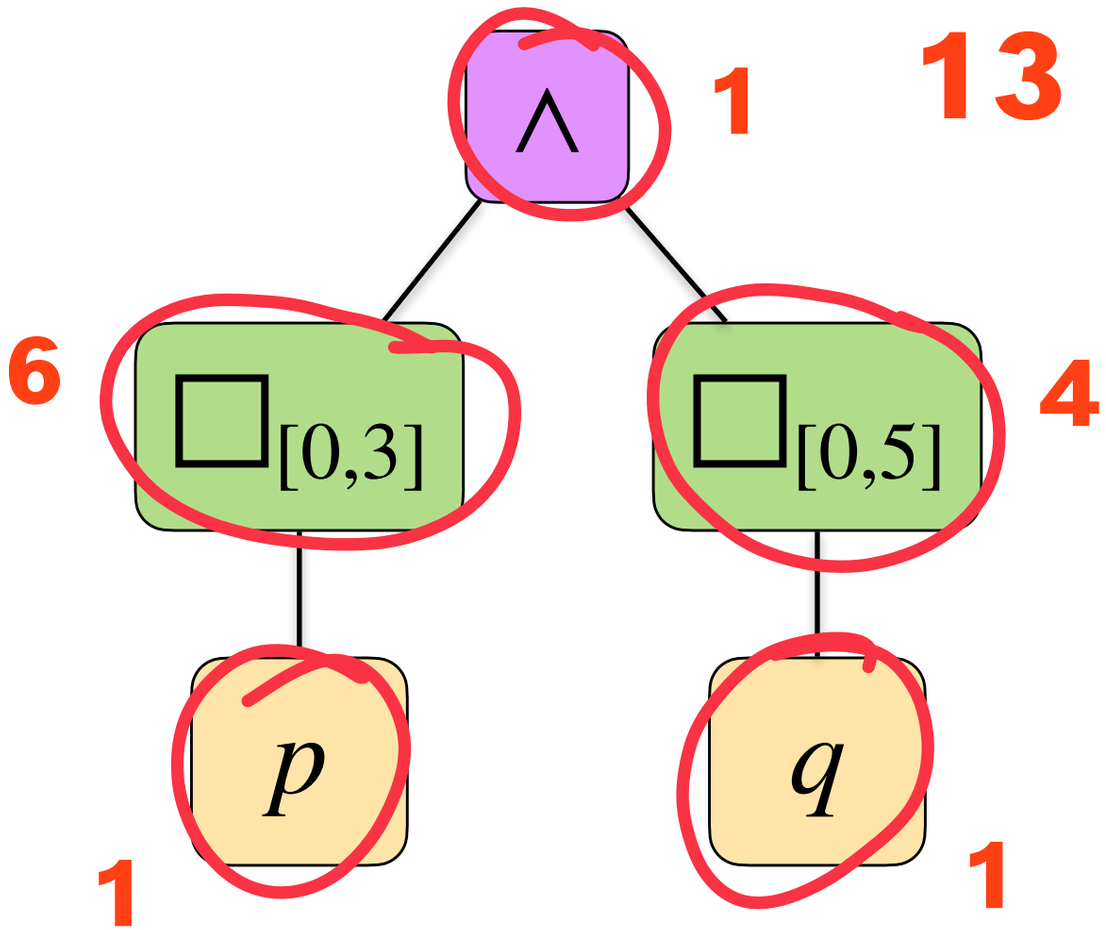
$$\square_{[0,3]} p \wedge \square_{[0,5]} q \quad \Rightarrow \quad \square_{[0,3]} (p \wedge \square_{[0,2]} q)$$

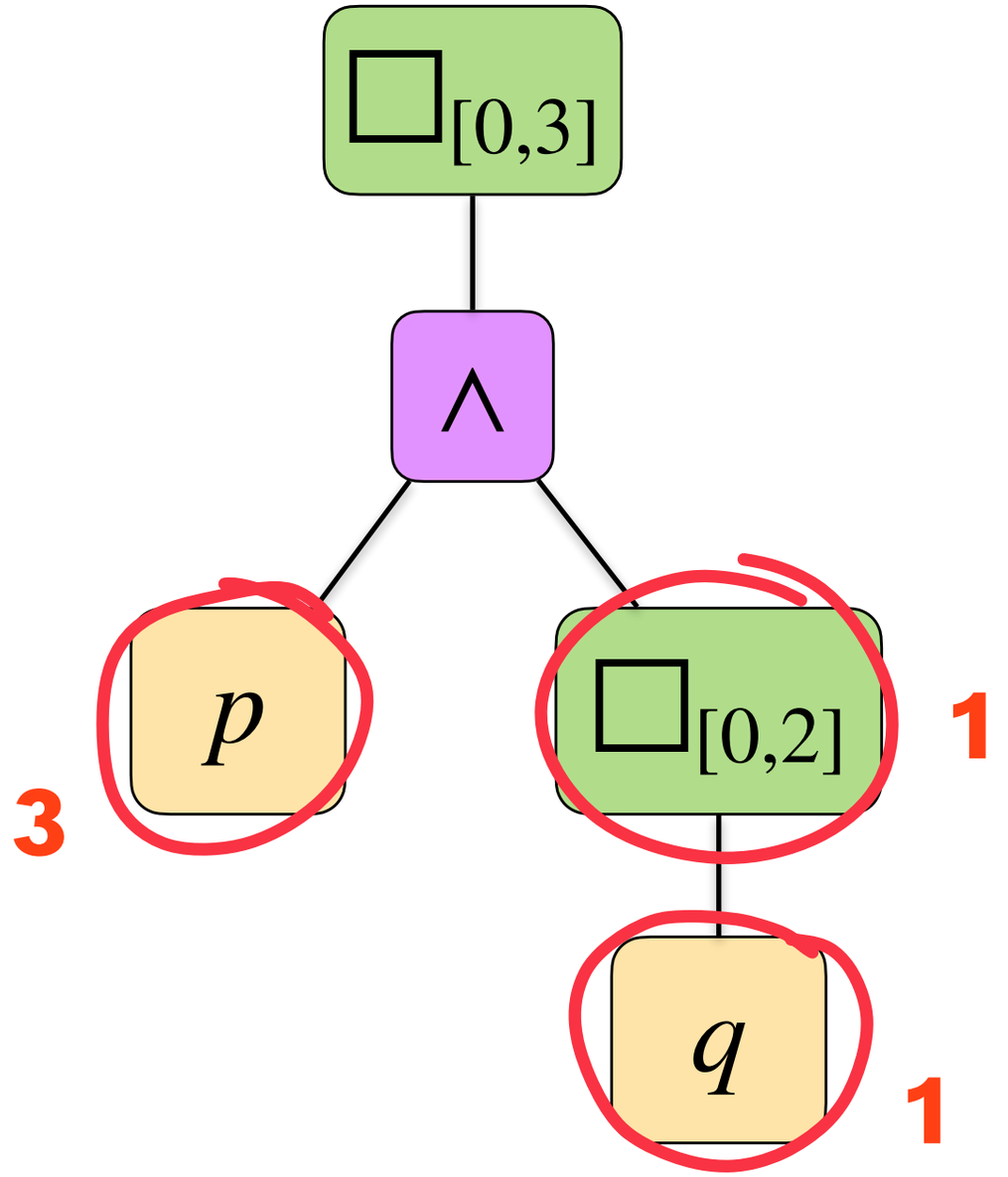




MLTL Rewrite Rules

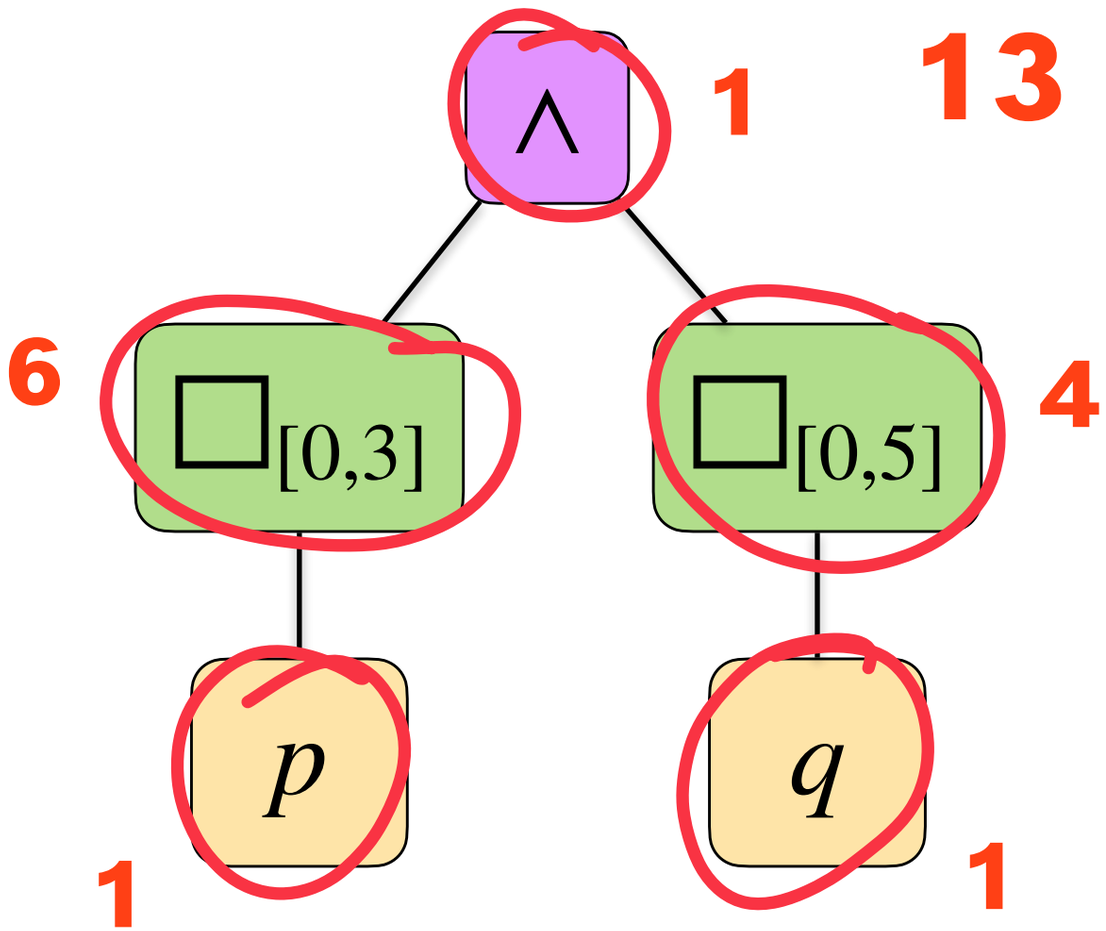
$$\square_{[0,3]} p \wedge \square_{[0,5]} q \quad \Rightarrow \quad \square_{[0,3]} (p \wedge \square_{[0,2]} q)$$

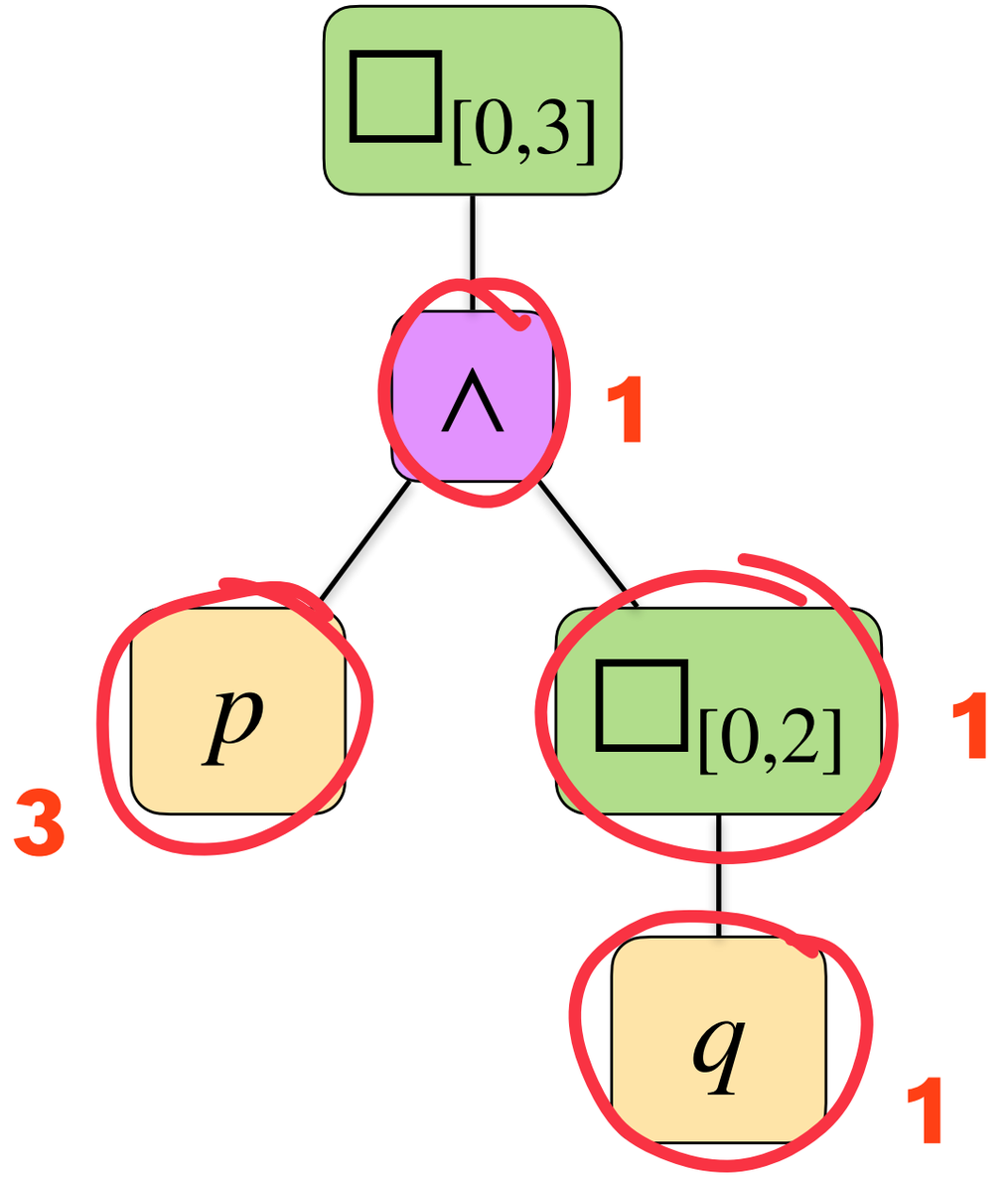




MLTL Rewrite Rules

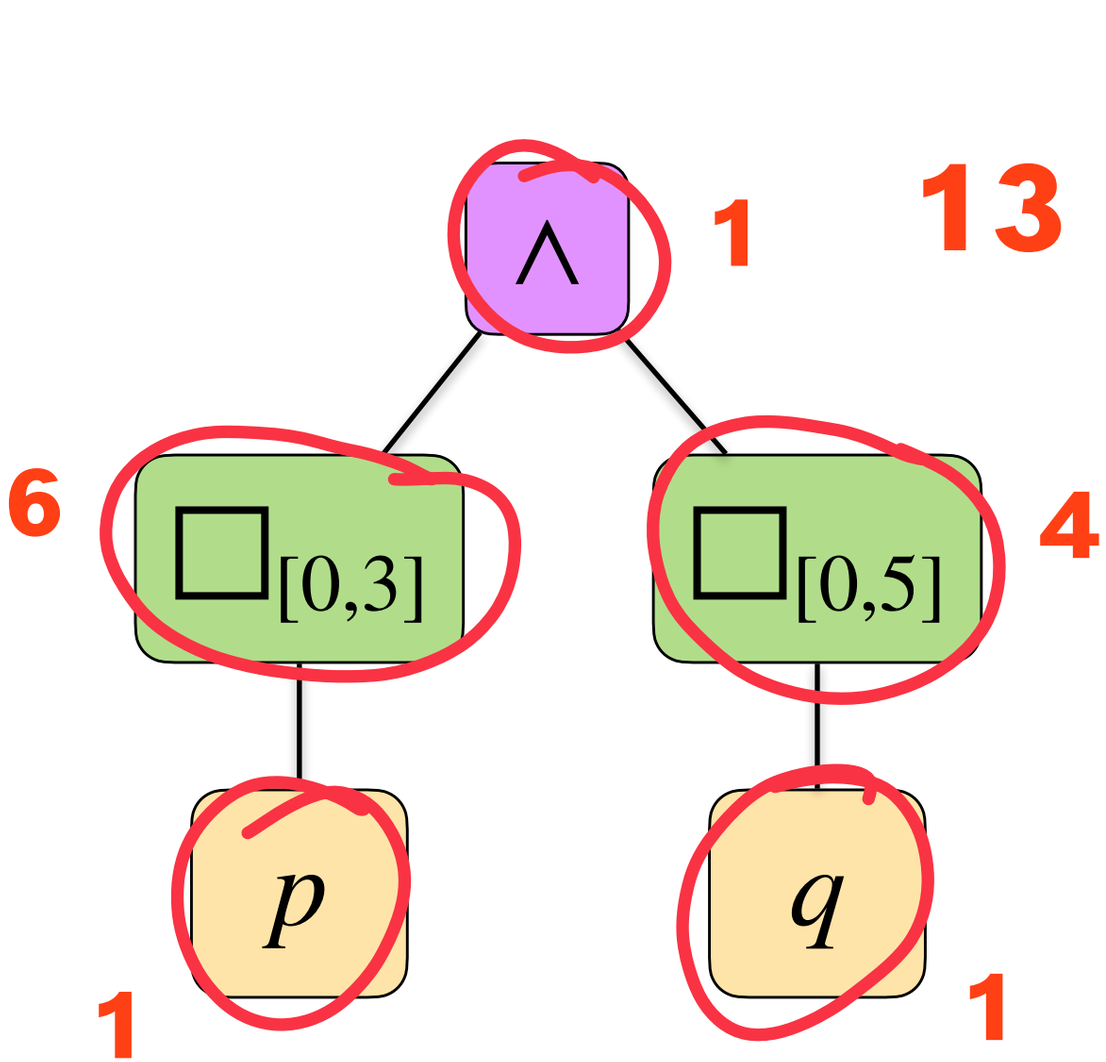
$$\square_{[0,3]} p \wedge \square_{[0,5]} q \quad \Rightarrow \quad \square_{[0,3]} (p \wedge \square_{[0,2]} q)$$

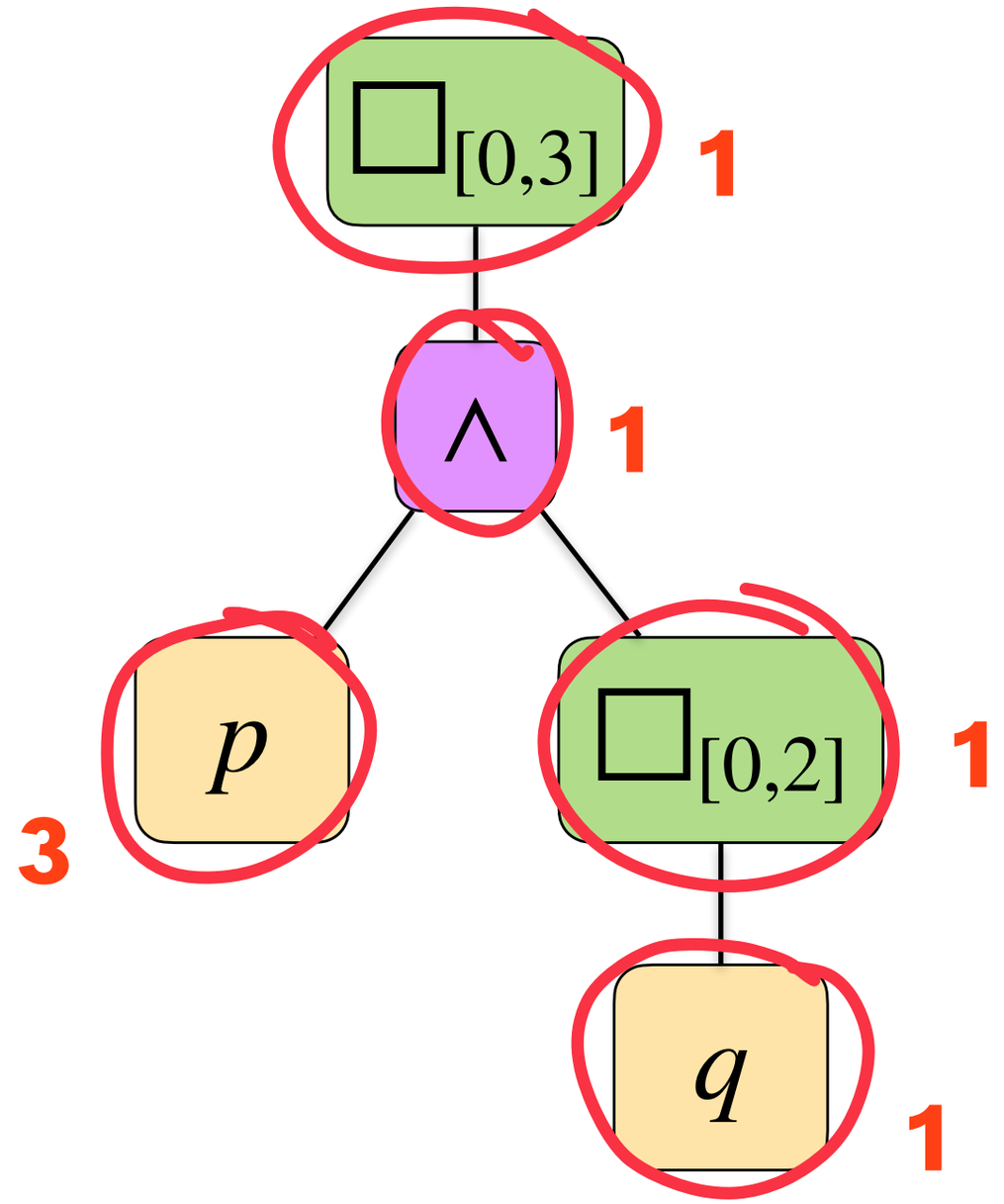




MLTL Rewrite Rules

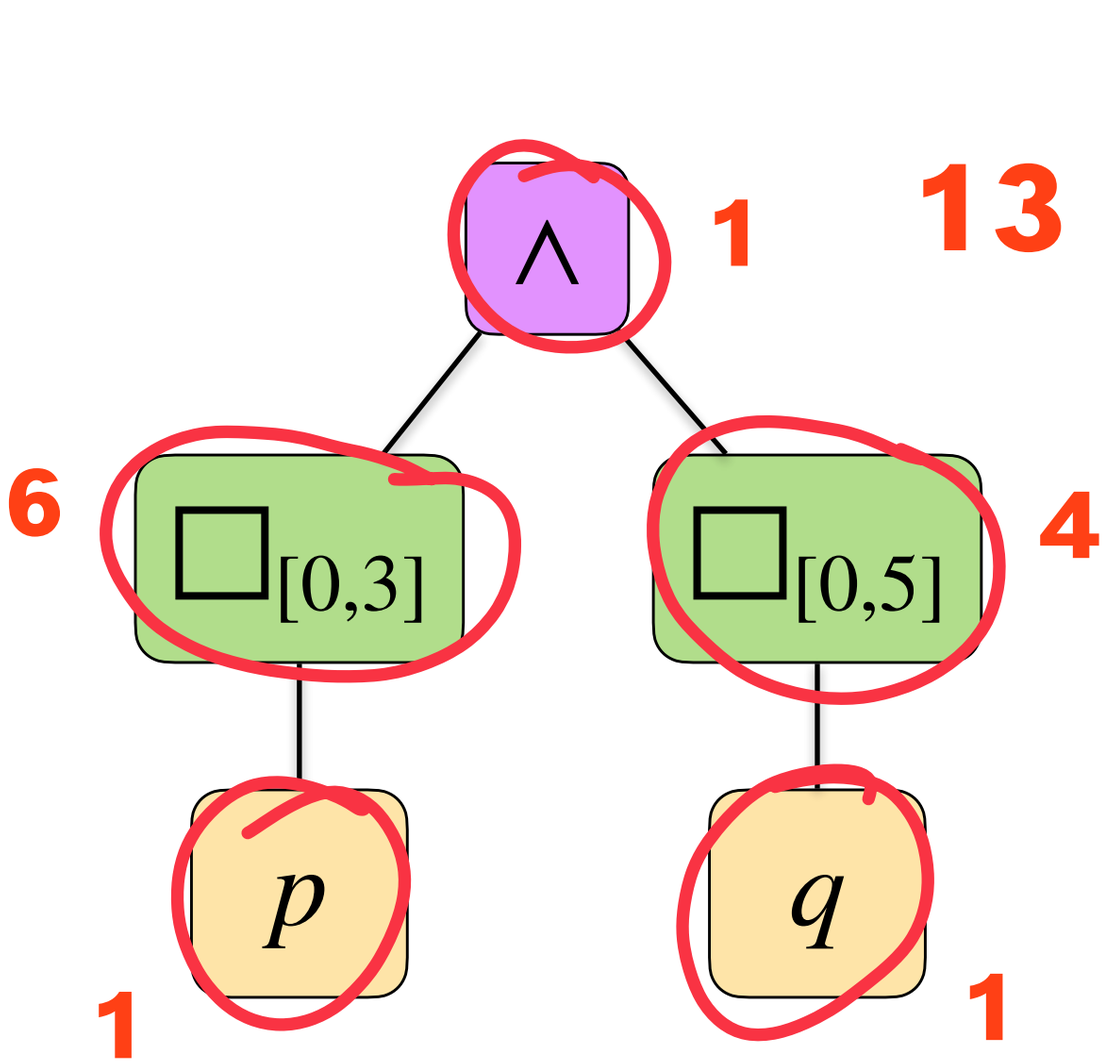
$$\square_{[0,3]} p \wedge \square_{[0,5]} q \quad \Rightarrow \quad \square_{[0,3]} (p \wedge \square_{[0,2]} q)$$

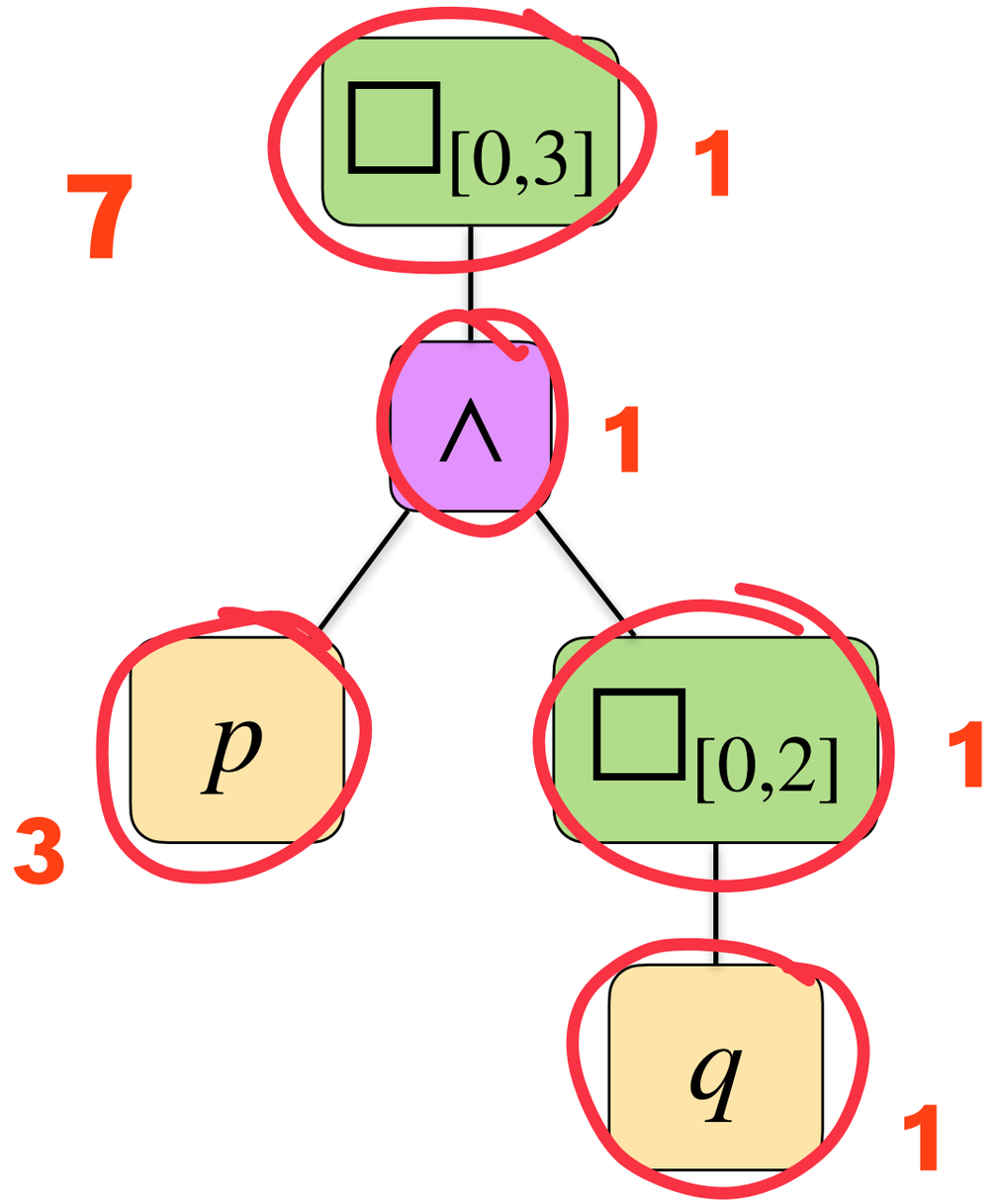




MLTL Rewrite Rules

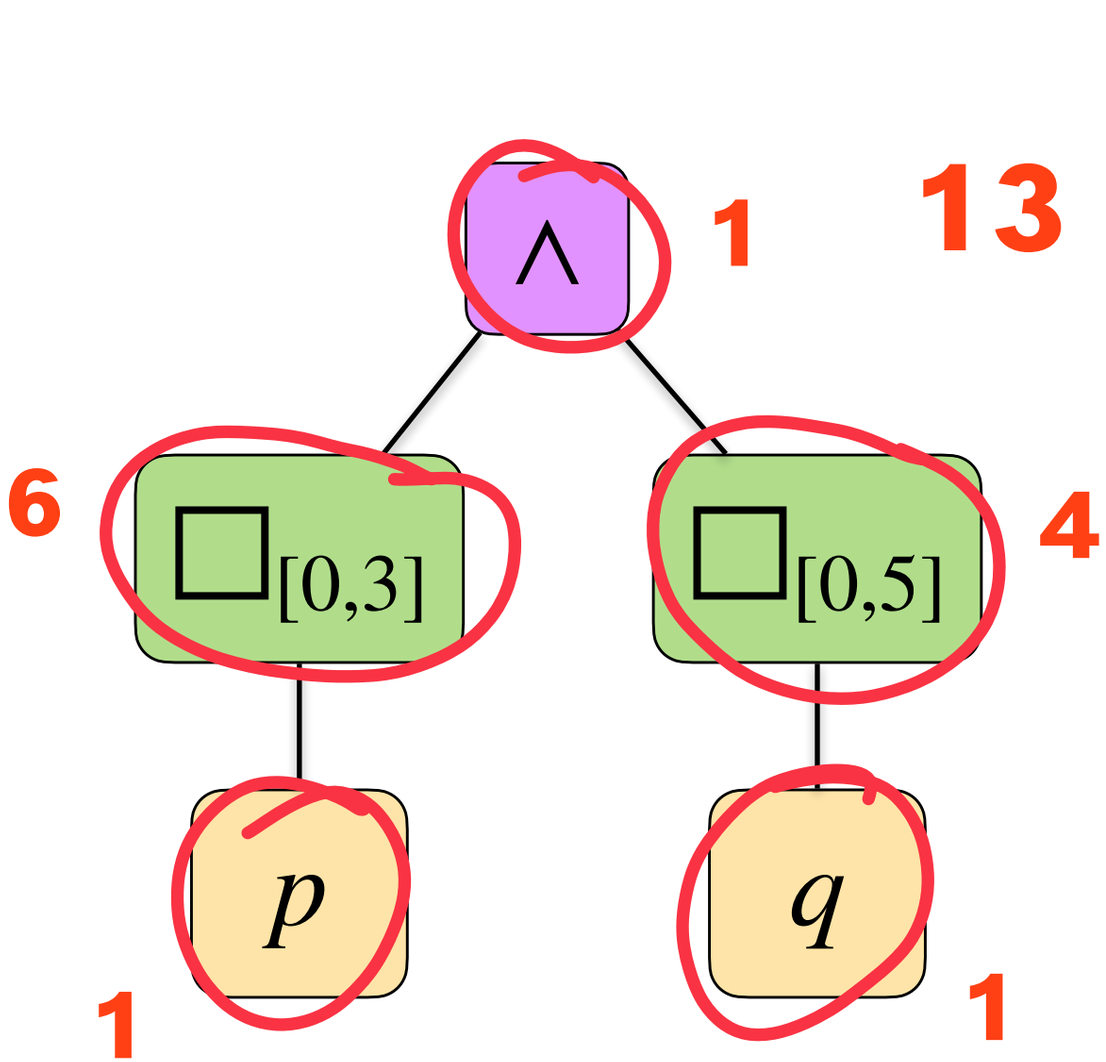
$$\square_{[0,3]} p \wedge \square_{[0,5]} q \quad \Rightarrow \quad \square_{[0,3]} (p \wedge \square_{[0,2]} q)$$



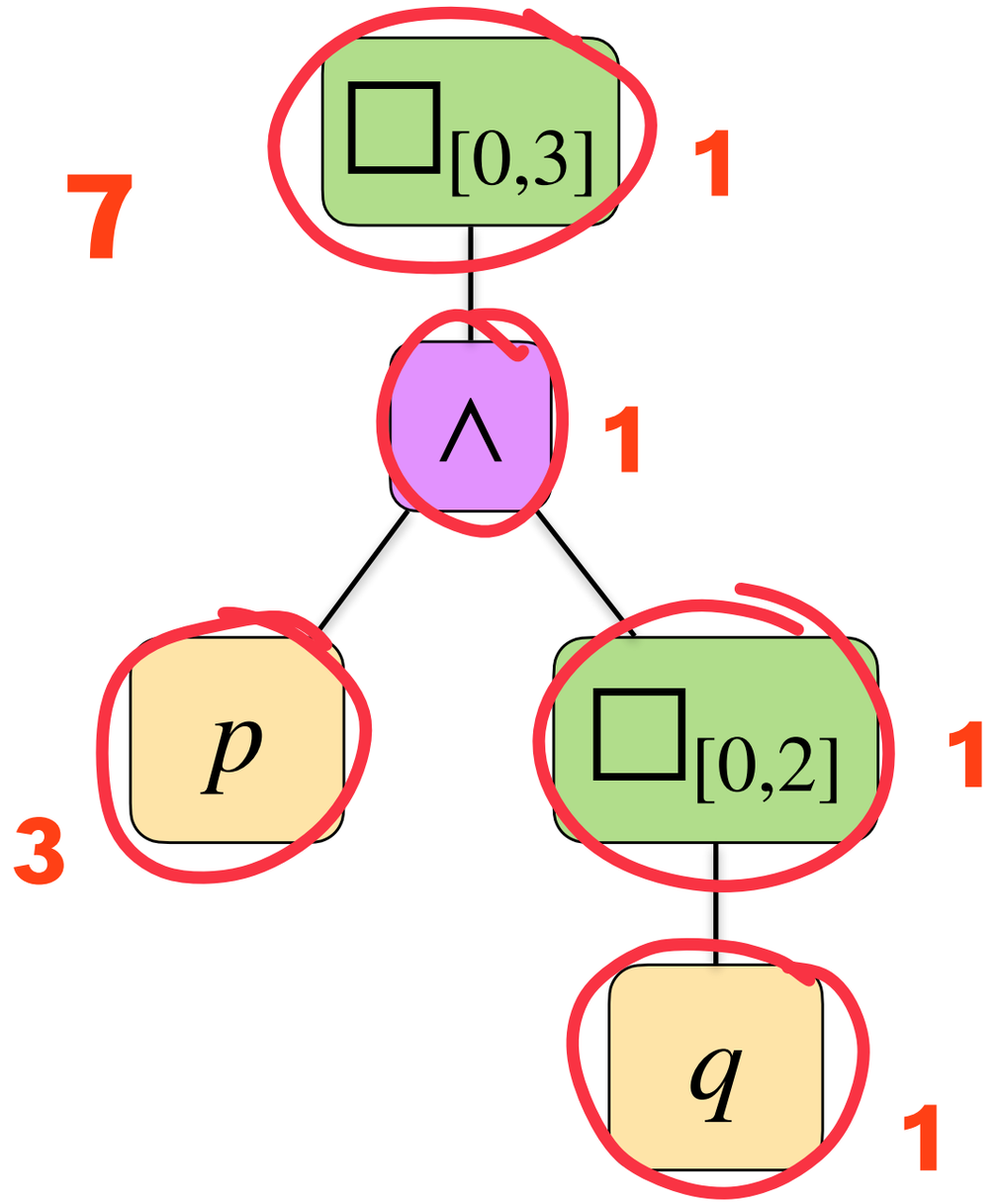


MLTL Rewrite Rules

$$\square_{[0,3]} p \wedge \square_{[0,5]} q \quad \Rightarrow \quad \square_{[0,3]} (p \wedge \square_{[0,2]} q)$$



13 > 7



MLTL Rewrite Rules [19]

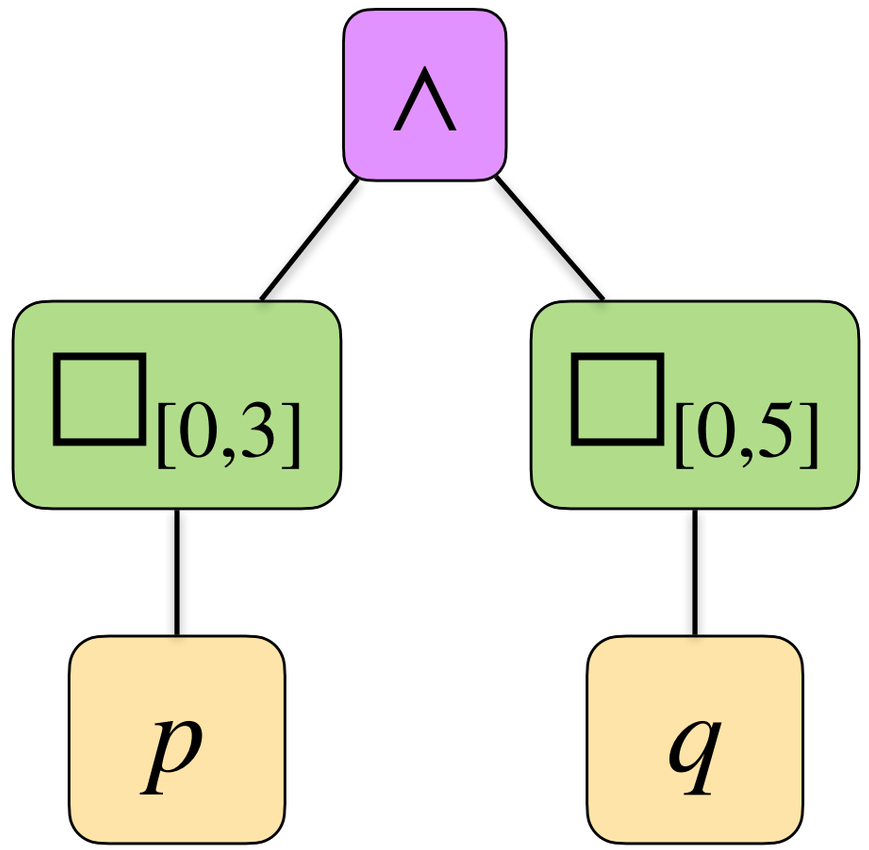
$\Box_{[l_1,u_1]}\Box_{[l_2,u_2]}\varphi \mapsto \Box_{[l_1+l_2,u_1+u_2]}\varphi$	$\Diamond_{[l_1,u_1]}\Diamond_{[l_2,u_2]}\varphi \mapsto \Diamond_{[l_1+l_2,u_1+u_2]}\varphi$	(R1)
$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\psi \mapsto \Box_{[l_3,u_3]}(\Box_{[l_1-l_3,u_1-u_3]}\varphi \wedge \Box_{[l_2-l_3,u_2-u_3]}\psi)$		
$\Diamond_{[l_1,u_1]}\varphi \vee \Diamond_{[l_2,u_2]}\psi \mapsto \Diamond_{[l_3,u_3]}(\Diamond_{[l_1-l_3,u_1-u_3]}\varphi \vee \Diamond_{[l_2-l_3,u_2-u_3]}\psi)$		(R2)
where $l_3 = min(l_1, l_2), u_3 = l_3 + min(u_1 - l_1, u_2 - l_2), l_3 < u_3$		
$\Box_{[a,a]}\Diamond_{[l,u]}\varphi \mapsto \Diamond_{[l+a,u+a]}\varphi$	$\Diamond_{[l,u]}\Box_{[a,a]}\varphi \mapsto \Diamond_{[l+a,u+a]}\varphi$	(R3)
$\Diamond_{[a,a]}\Box_{[l,u]}\varphi \mapsto \Box_{[l+a,u+a]}\varphi$	$\Box_{[l,u]}\Diamond_{[a,a]}\varphi \mapsto \Box_{[l+a,u+a]}\varphi$	
$\Box_{[l_1,u_1]}\varphi \wedge \Box_{[l_2,u_2]}\varphi \mapsto \Box_{[l_1,u_3]}\varphi$	$\Diamond_{[l_1,u_1]}\varphi \vee \Diamond_{[l_2,u_2]}\varphi \mapsto \Diamond_{[l_1,u_2]}\varphi$	(R4)
where $l_1 \leq l_2 \leq u_1 + 1, u_3 = max(u_1, u_2)$		
$\Box_{[l_1,u_1]}\varphi \vee \Box_{[l_2,u_2]}\varphi \mapsto \Box_{[l_2,u_2]}\varphi$	$\Diamond_{[l_1,u_1]}\varphi \wedge \Diamond_{[l_2,u_2]}\varphi \mapsto \Diamond_{[l_2,u_2]}\varphi$	(R5)
where $l_1 \leq l_2 \leq u_2 \leq u_1$		
$\Box_{[a,a]}(\varphi\ \mathcal{U}_{[l,u]}\ \psi) \mapsto \varphi\ \mathcal{U}_{[l+a,u+a]}\ \psi$	$(\Box_{[a,a]}\varphi)\ \mathcal{U}_{[l,u]}\ (\Box_{[a,a]}\psi) \mapsto \varphi\ \mathcal{U}_{[l+a,u+a]}\ \psi$	(R6)
$(\varphi_1\ \mathcal{U}_{[l,u_1]}\ \varphi_2) \wedge (\varphi_3\ \mathcal{U}_{[l,u_2]}\ \varphi_2) \mapsto (\varphi_1 \wedge \varphi_3)\ \mathcal{U}_{[l,u_1]}\ \varphi_2$		(R7)
where $l \leq u_1, l \leq u_2, u_1 \leq u_2$		
$\varphi\ \mathcal{U}_{[l_1,u_1]}\ \Box_{[0,u_2]}\varphi \mapsto \Box_{[l_1,l_1+u_2]}\varphi$	$\varphi\ \mathcal{U}_{[l_1,u_1]}\ \Diamond_{[0,u_2]}\varphi \mapsto \Diamond_{[l_1,l_1+u_2]}\varphi$	(R8)

How to apply rewrite rules algorithmically?

- **Order of rewrites matter** — applying one rule may **disable** the application of another.
  - Assume R_1 disables R_3
  - $R_1 \longrightarrow R_2$
  - $R_2 \longrightarrow R_3 \longrightarrow R_1$
  - ...
- Classic problem in **program optimization [20]**
- Solution: **Equality Saturation [21]**

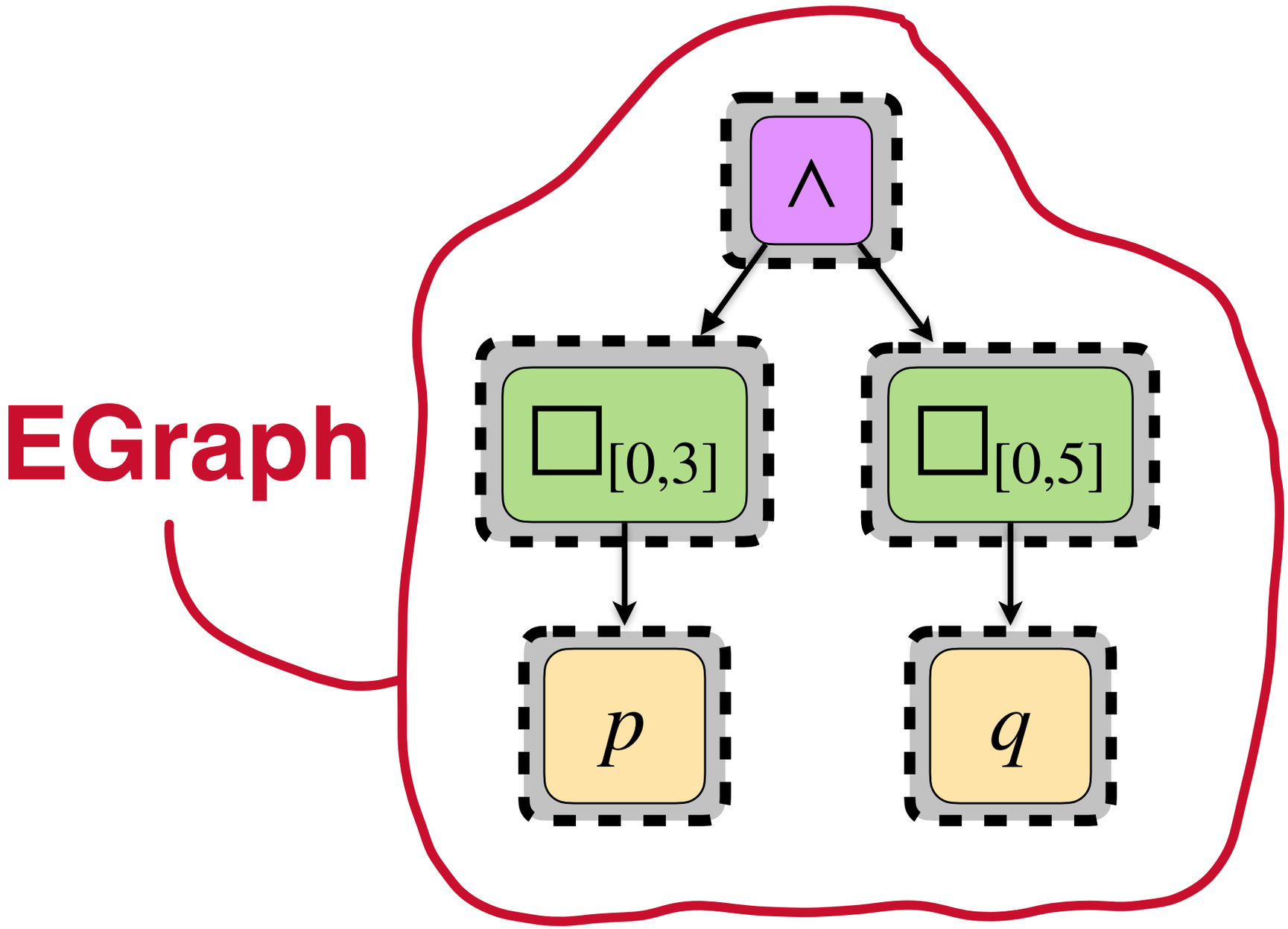
MLTL Equality Saturation

$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$



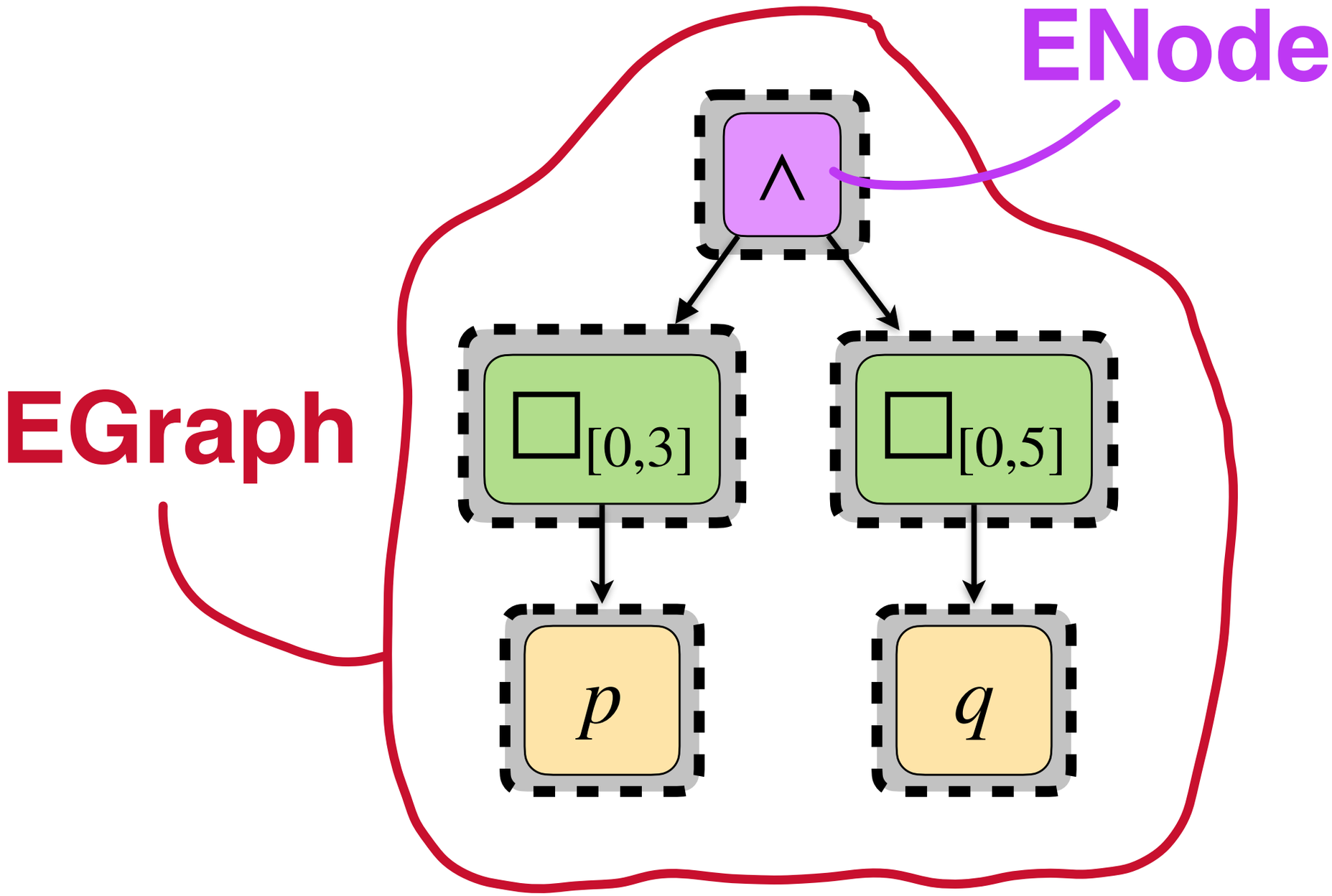
MLTL Equality Saturation

$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$



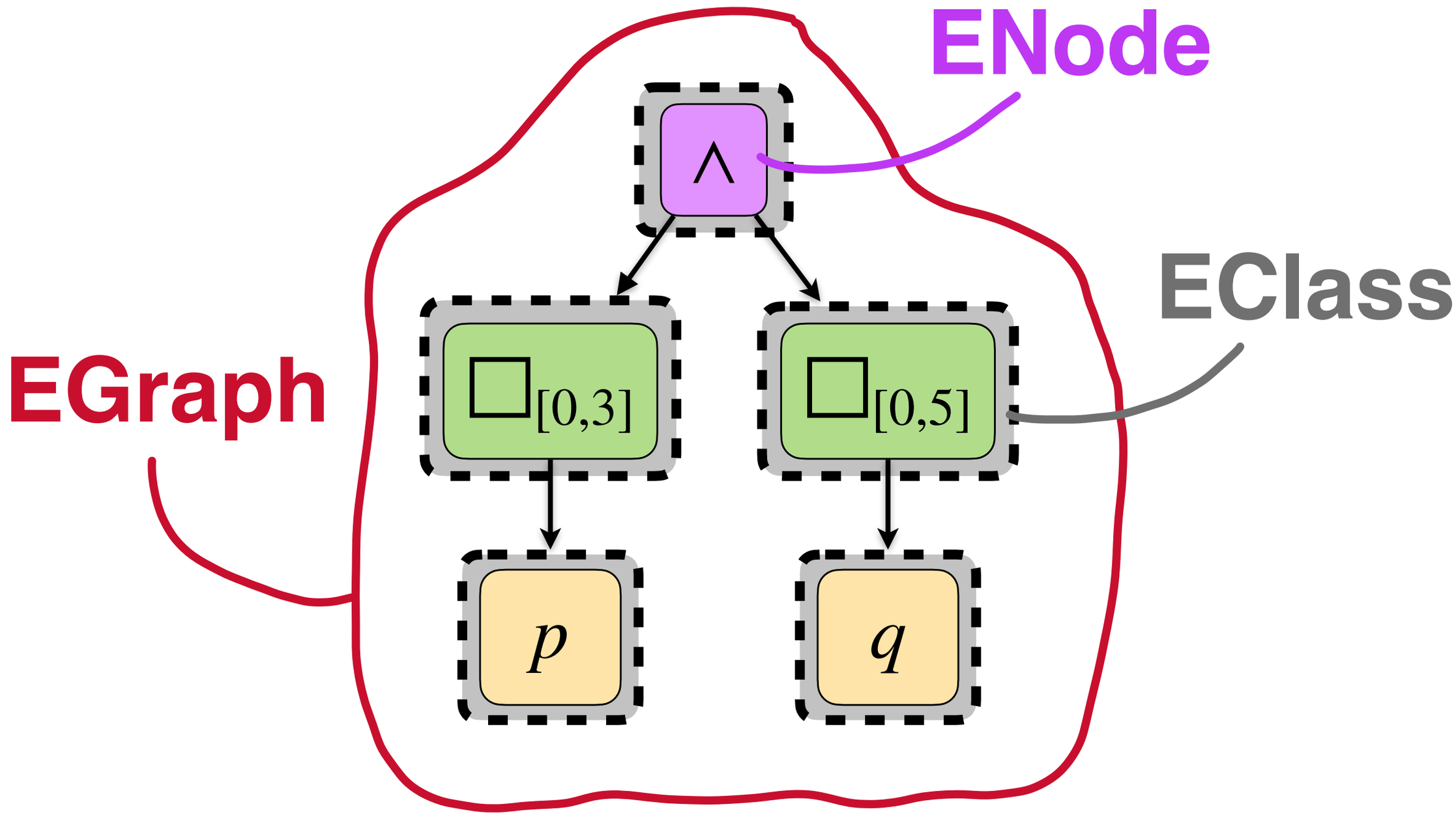
MLTL Equality Saturation

$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$



MLTL Equality Saturation

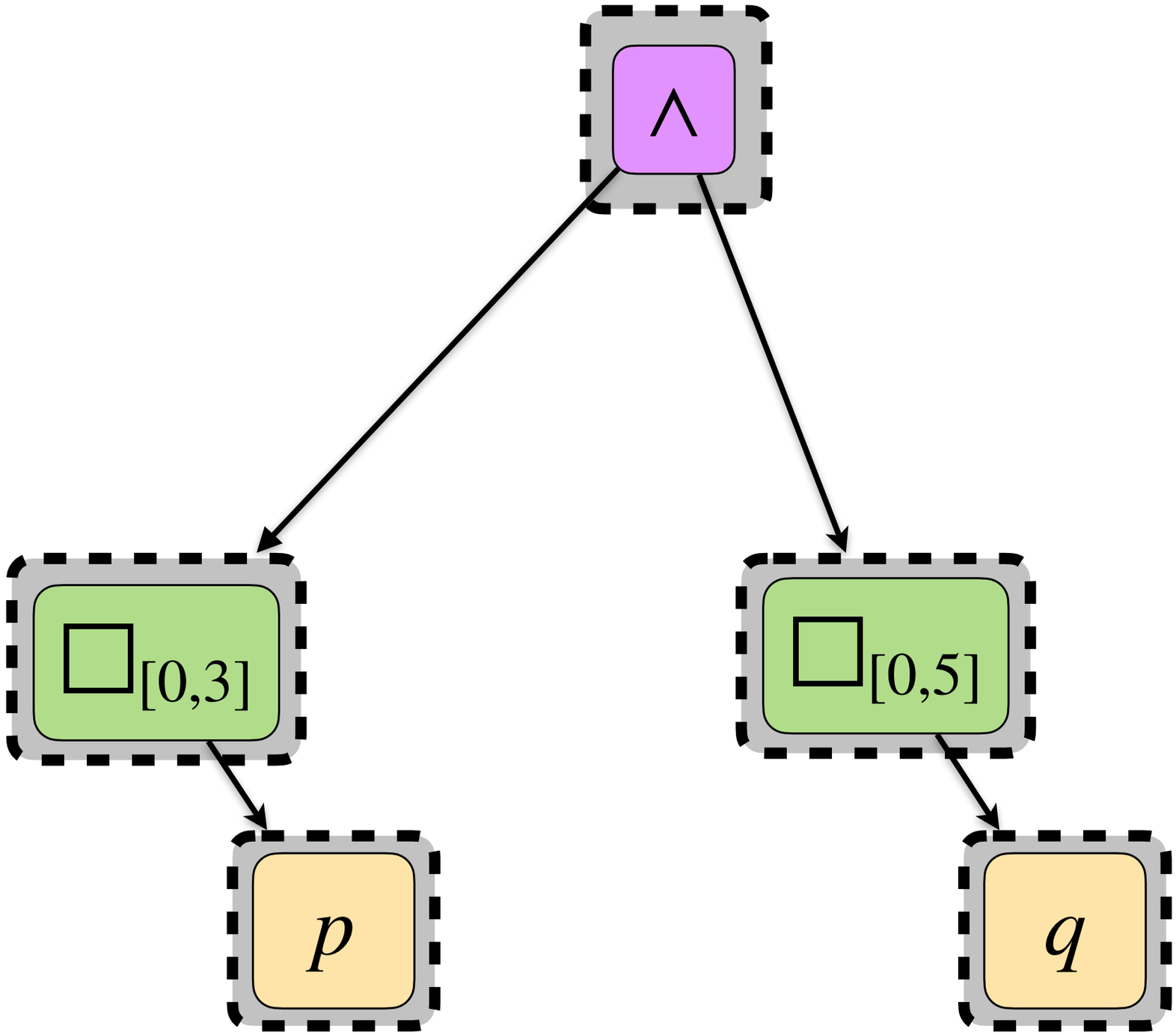
$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$



MLTL Equality Saturation

$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$

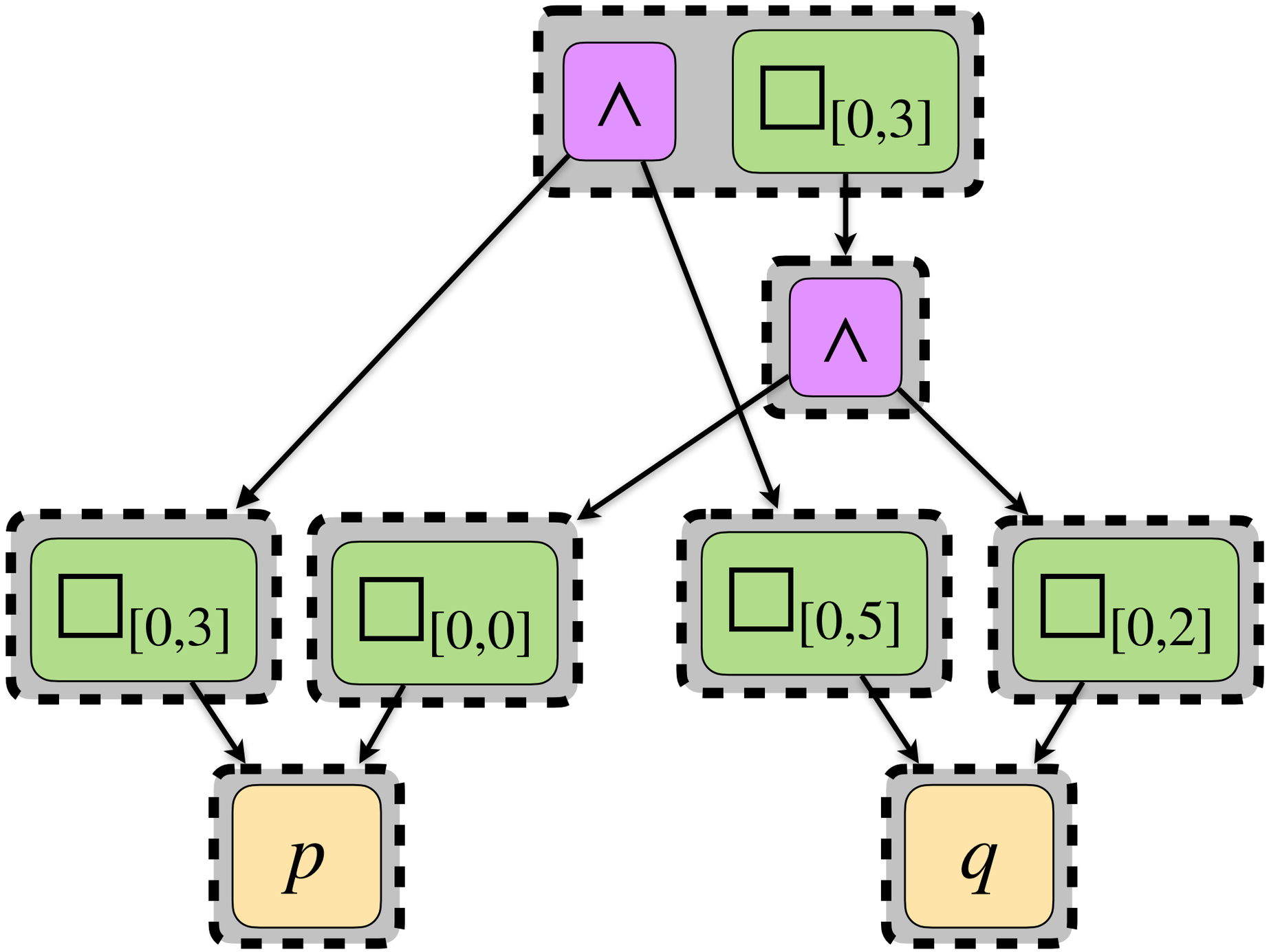
$$\Rightarrow \Box_{[0,3]} (\Box_{[0,0]} p \wedge \Box_{[0,2]} q)$$



MLTL Equality Saturation

$$\square_{[0,3]} p \wedge \square_{[0,5]} q$$

$$\Rightarrow \square_{[0,3]} (\square_{[0,0]} p \wedge \square_{[0,2]} q)$$

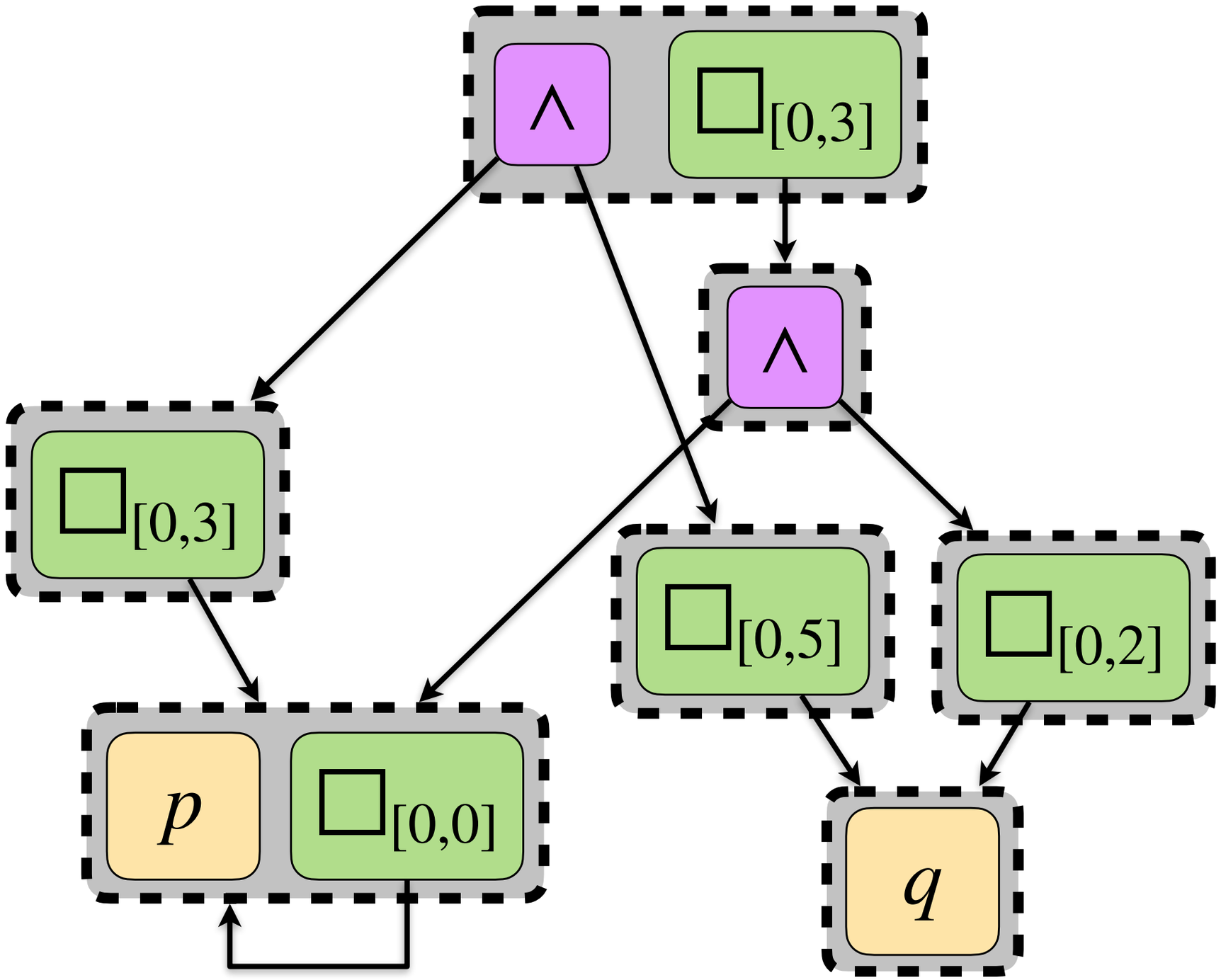


MLTL Equality Saturation

$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$

$$\Rightarrow \Box_{[0,3]} (\Box_{[0,0]} p \wedge \Box_{[0,2]} q)$$

$$\Rightarrow \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$



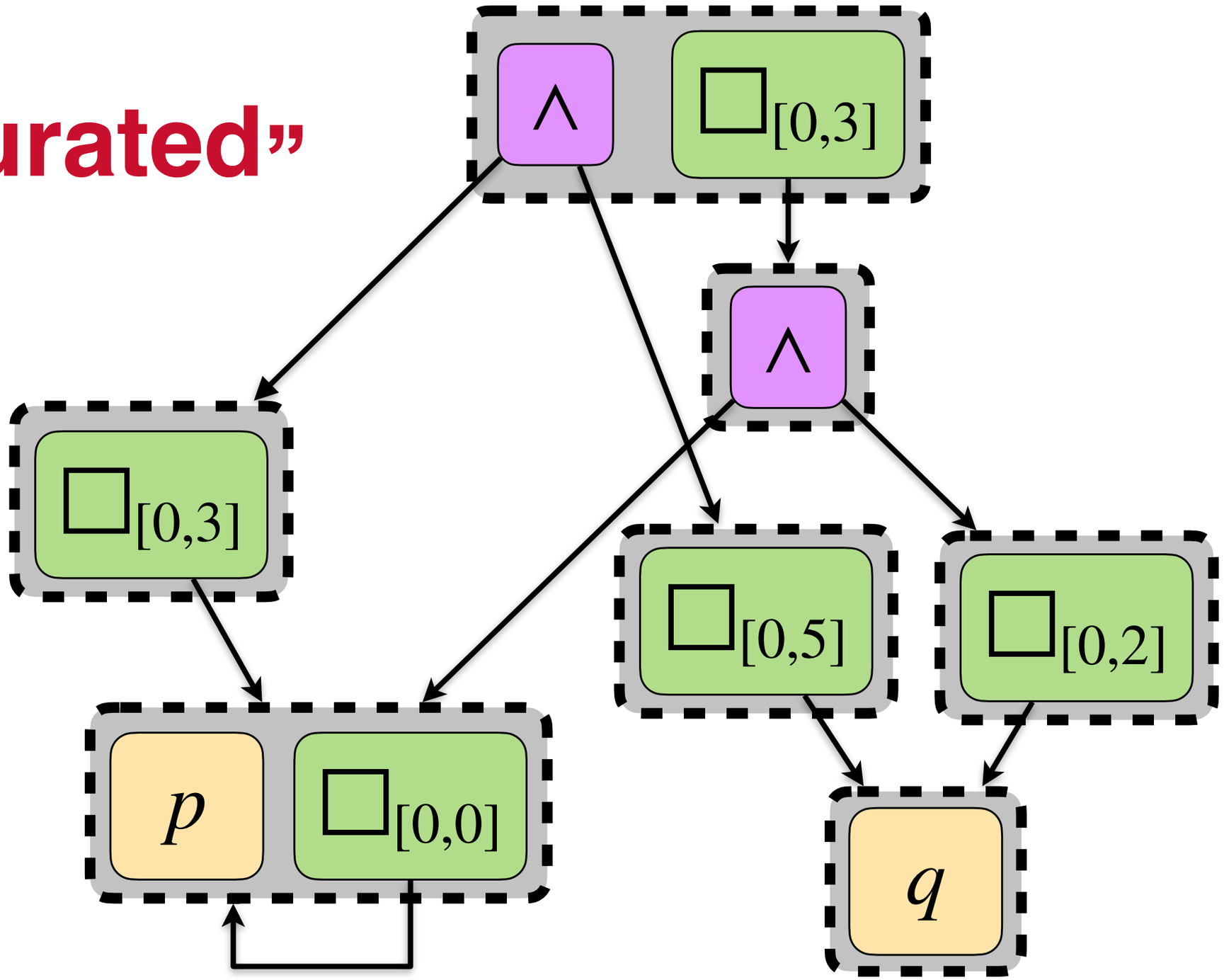
MLTL Equality Saturation

$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$

$$\Rightarrow \Box_{[0,3]} (\Box_{[0,0]} p \wedge \Box_{[0,2]} q)$$

$$\Rightarrow \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

“Saturated”

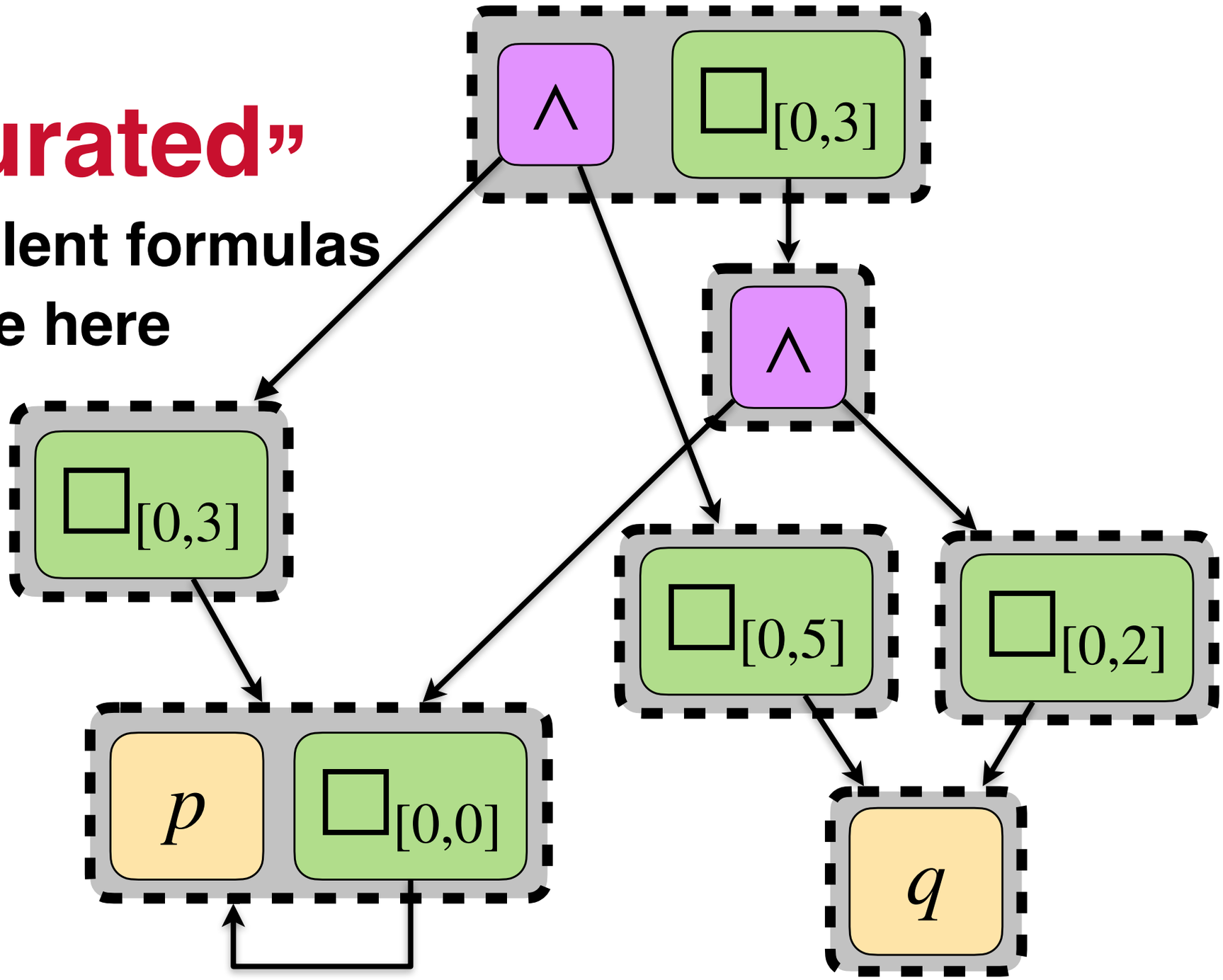


MLTL Equality Saturation

$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$
$$\Rightarrow \Box_{[0,3]} (\Box_{[0,0]} p \wedge \Box_{[0,2]} q)$$
$$\Rightarrow \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

“Saturated”

All equivalent formulas to orig. are here



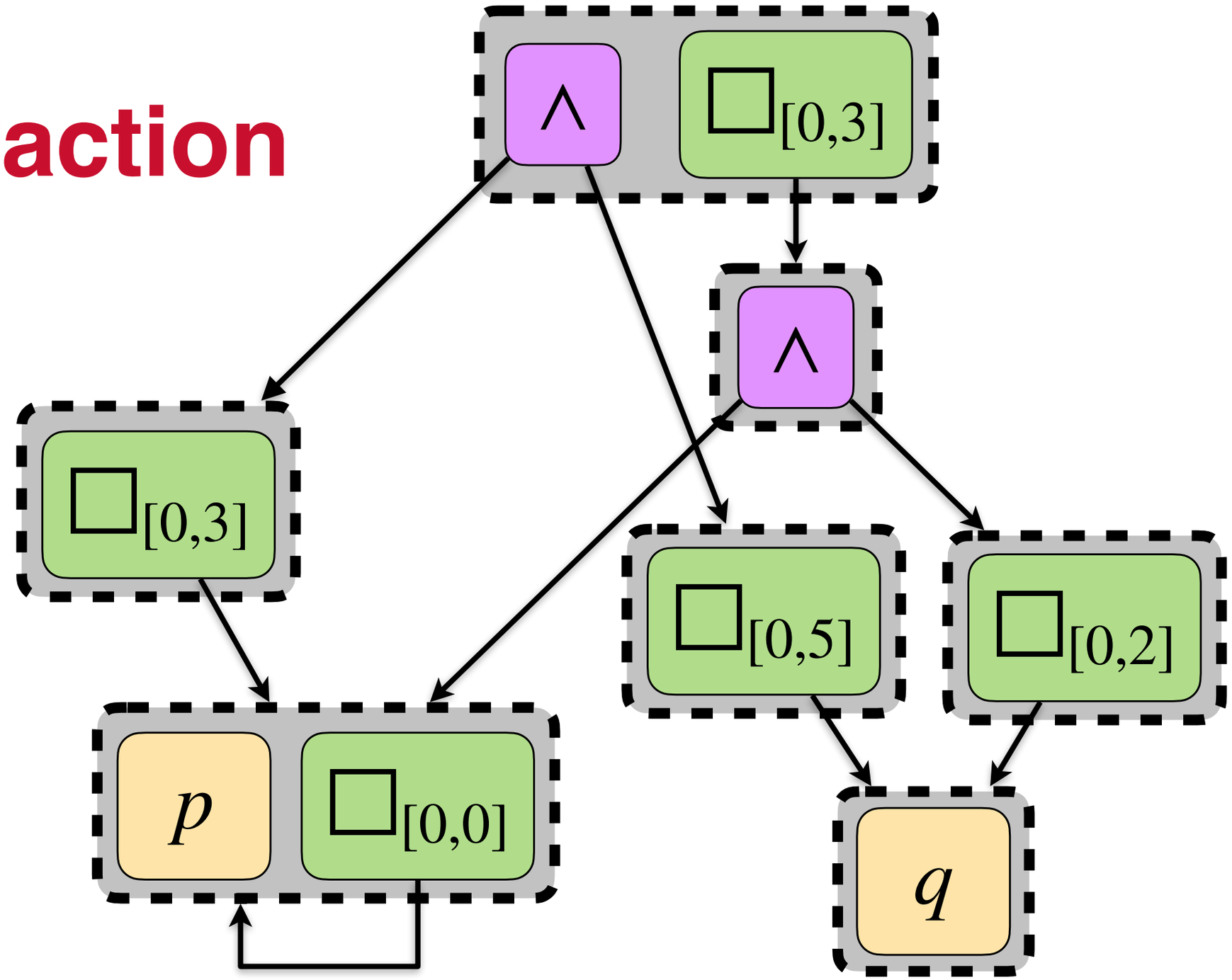
MLTL Equality Saturation

$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$

$$\Rightarrow \Box_{[0,3]} (\Box_{[0,0]} p \wedge \Box_{[0,2]} q)$$

$$\Rightarrow \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

Extraction

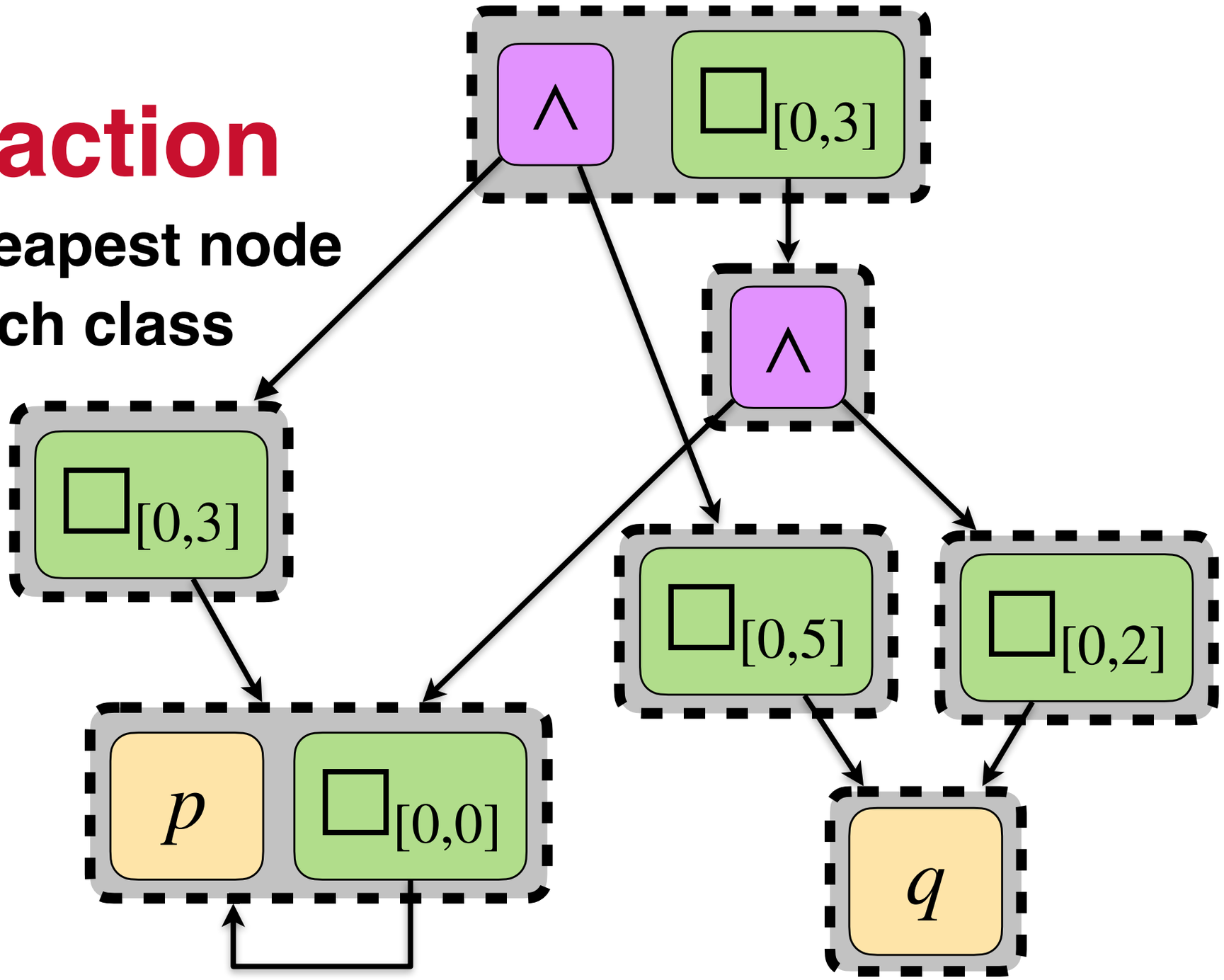


MLTL Equality Saturation

$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$
$$\Rightarrow \Box_{[0,3]} (\Box_{[0,0]} p \wedge \Box_{[0,2]} q)$$
$$\Rightarrow \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

Extraction

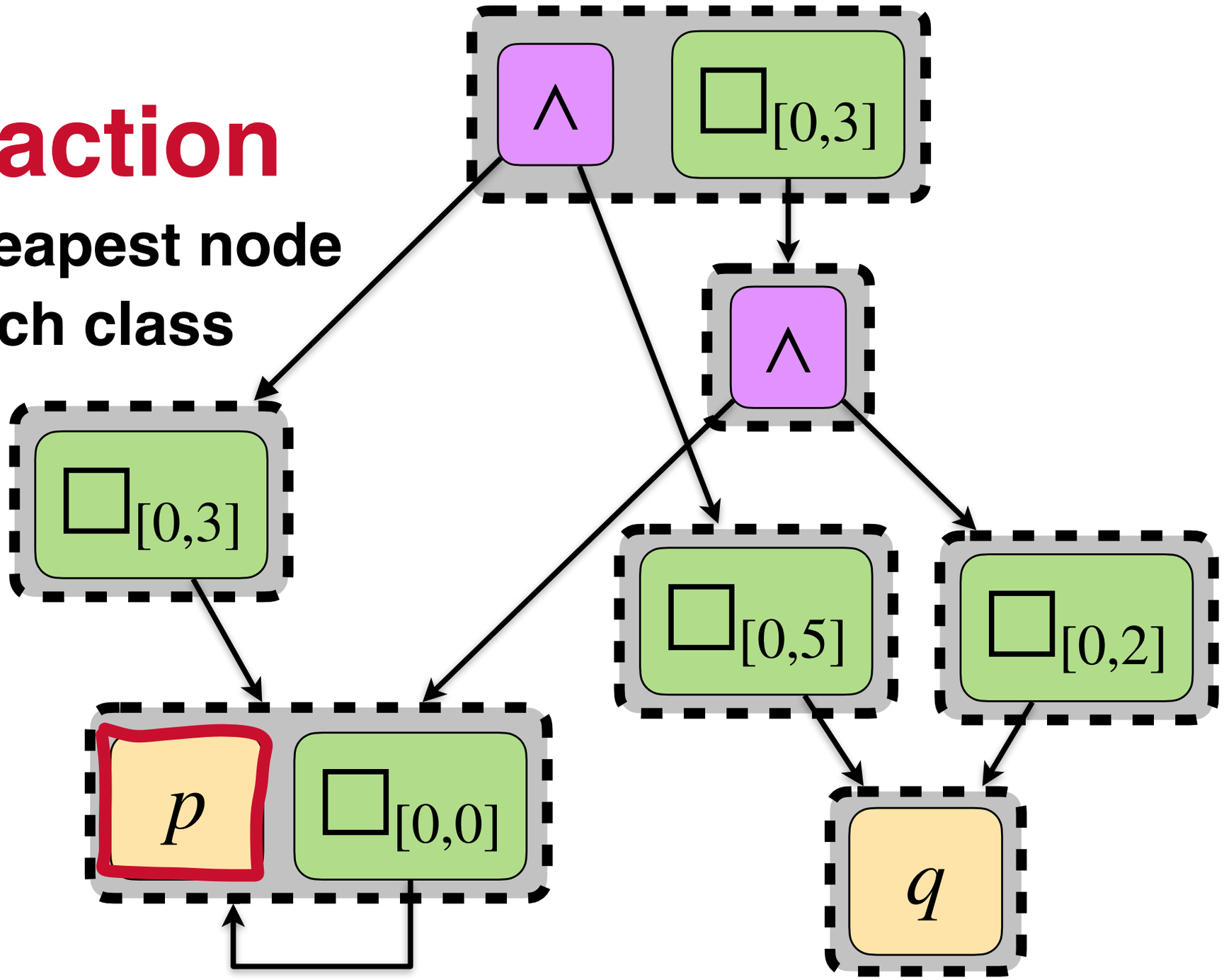
Pick cheapest node from each class



$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$
$$\Rightarrow \Box_{[0,3]} (\Box_{[0,0]} p \wedge \Box_{[0,2]} q)$$
$$\Rightarrow \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

Extraction

Pick cheapest node from each class

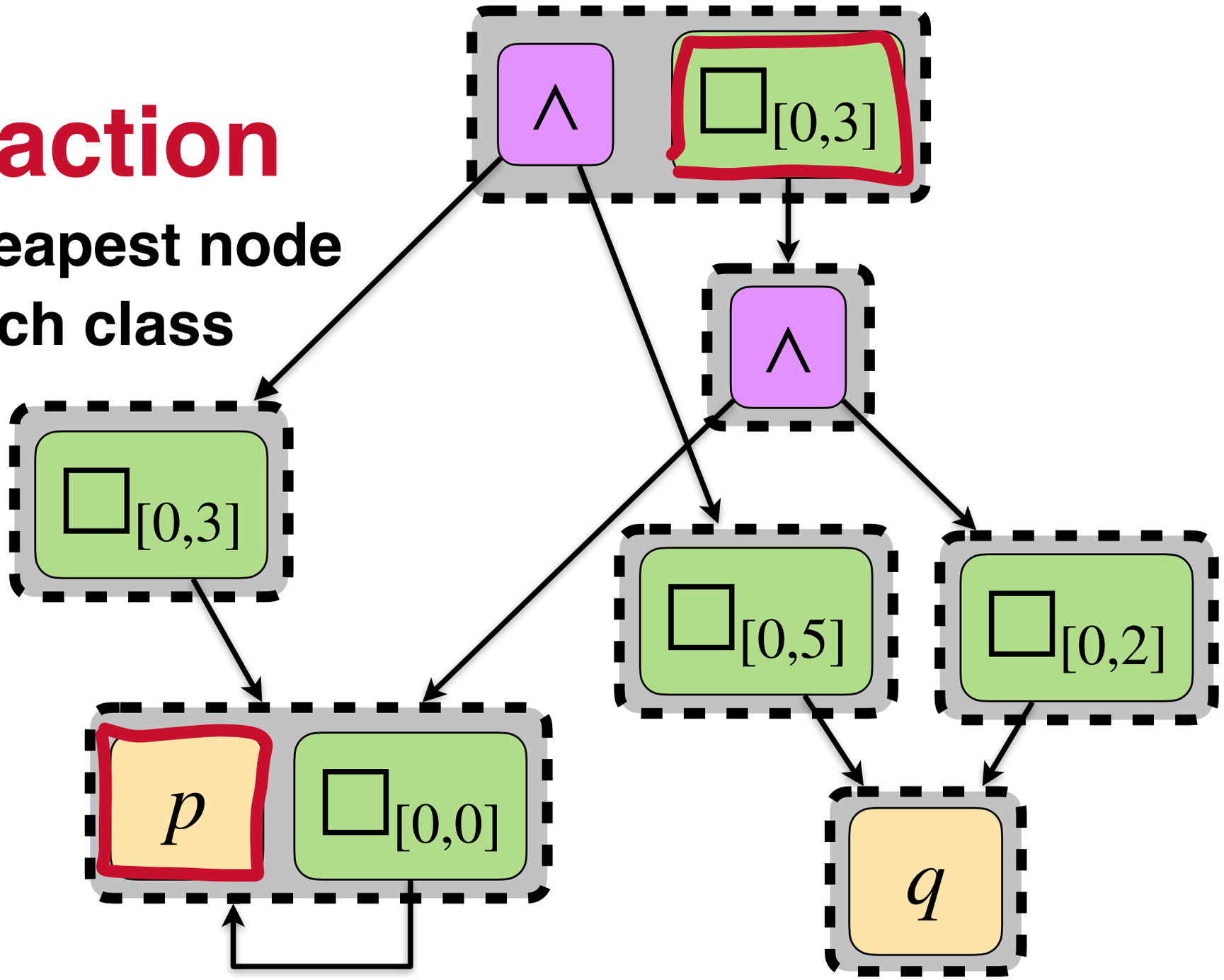


MLTL Equality Saturation

$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$
$$\Rightarrow \Box_{[0,3]} (\Box_{[0,0]} p \wedge \Box_{[0,2]} q)$$
$$\Rightarrow \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

Extraction

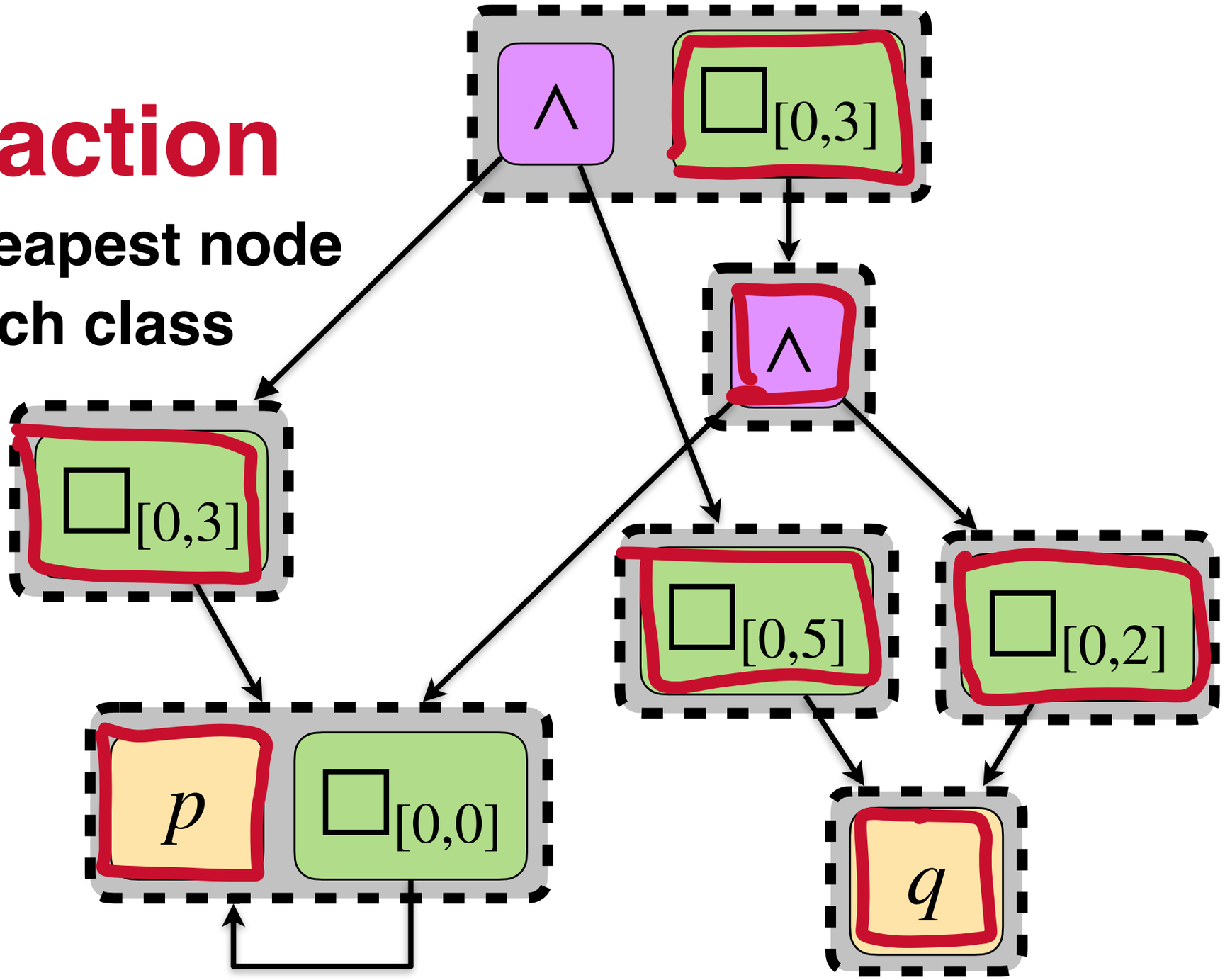
Pick cheapest node from each class



$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$
$$\Rightarrow \Box_{[0,3]} (\Box_{[0,0]} p \wedge \Box_{[0,2]} q)$$
$$\Rightarrow \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

Extraction

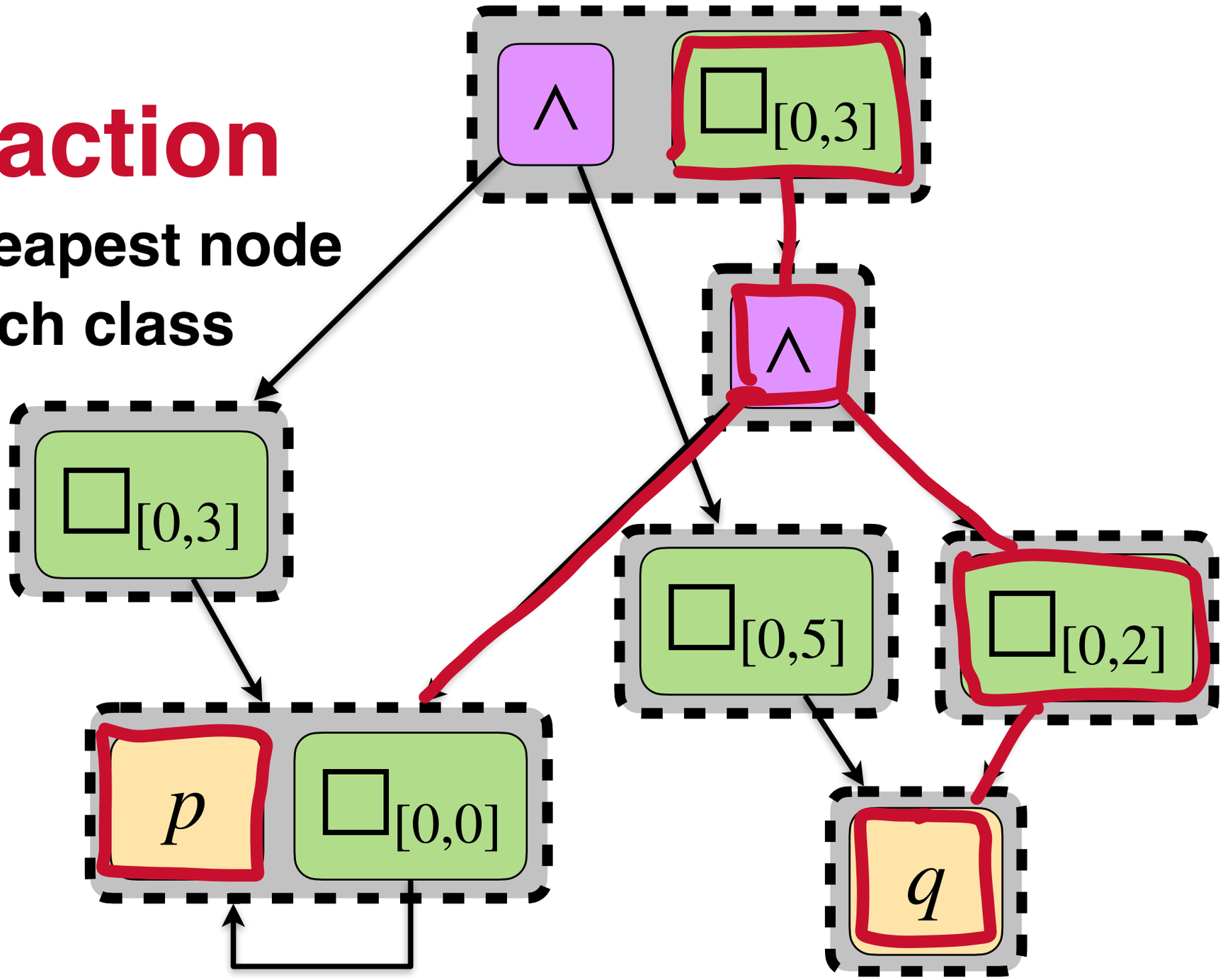
Pick cheapest node from each class



$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$
$$\Rightarrow \Box_{[0,3]} (\Box_{[0,0]} p \wedge \Box_{[0,2]} q)$$
$$\Rightarrow \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$

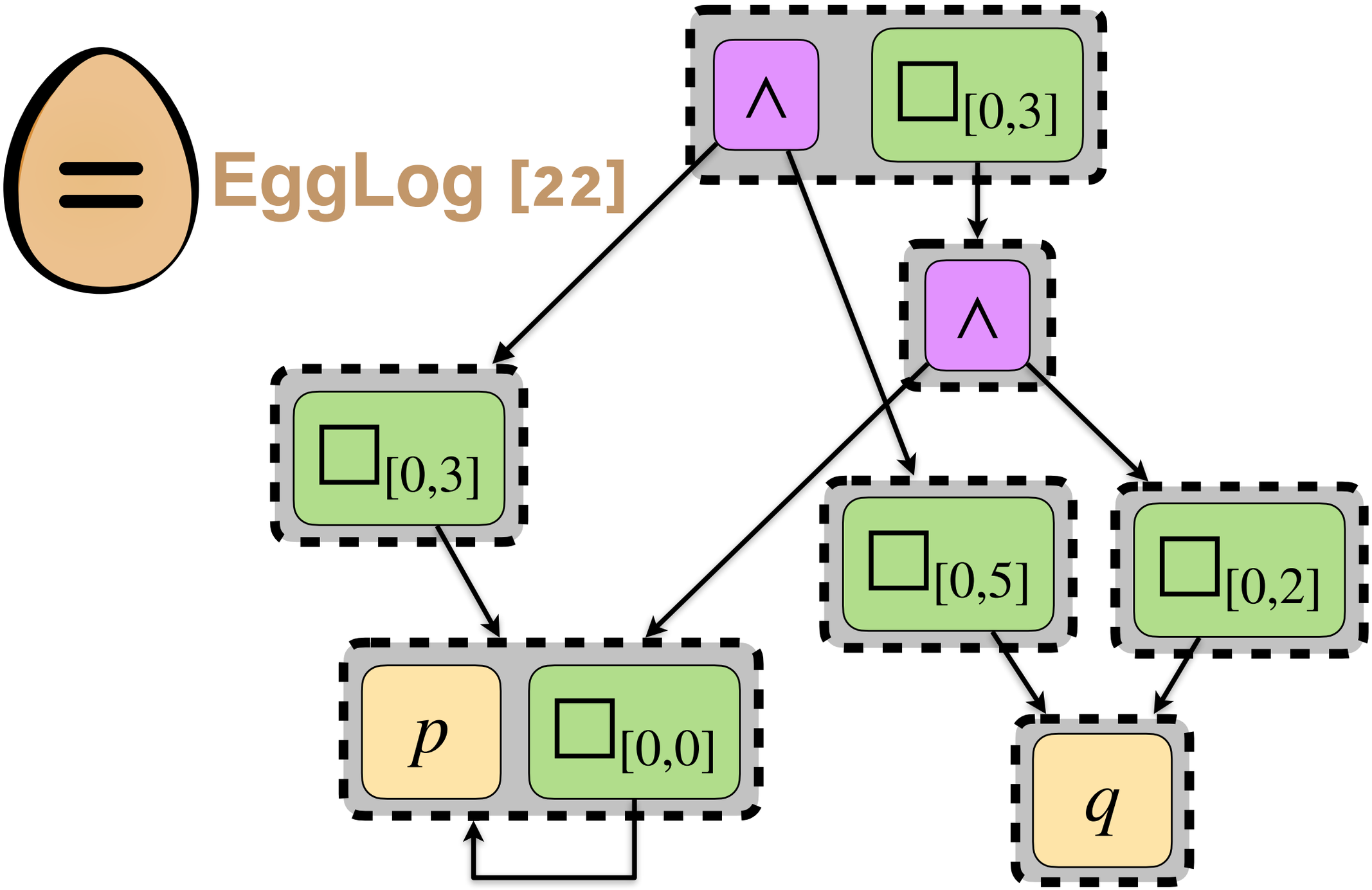
Extraction

Pick cheapest node from each class

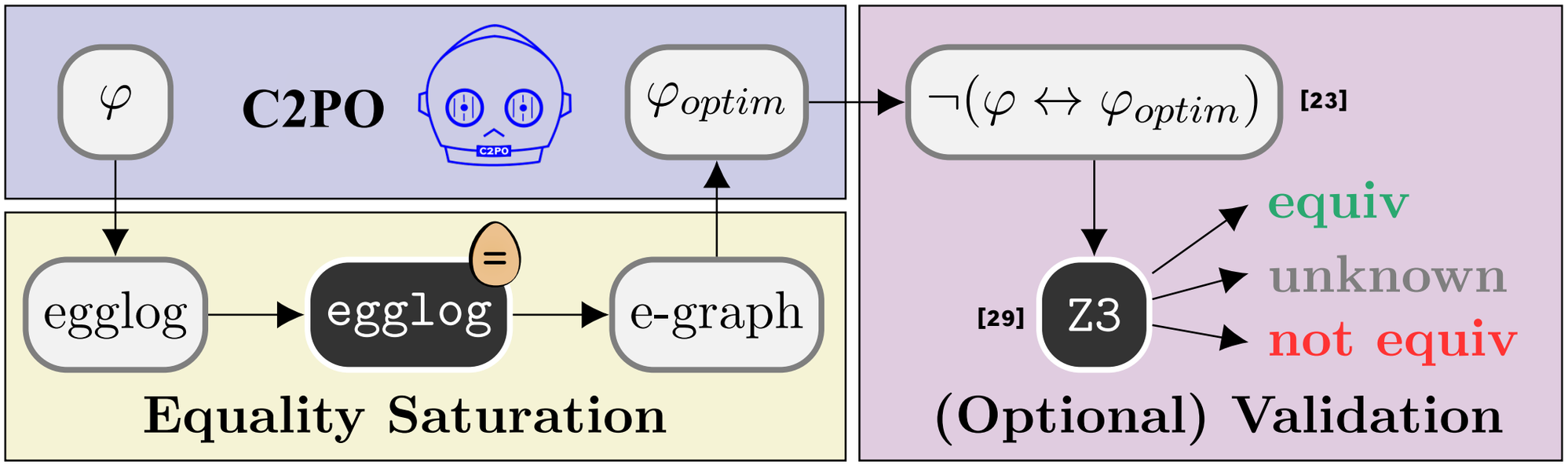


MLTL Equality Saturation

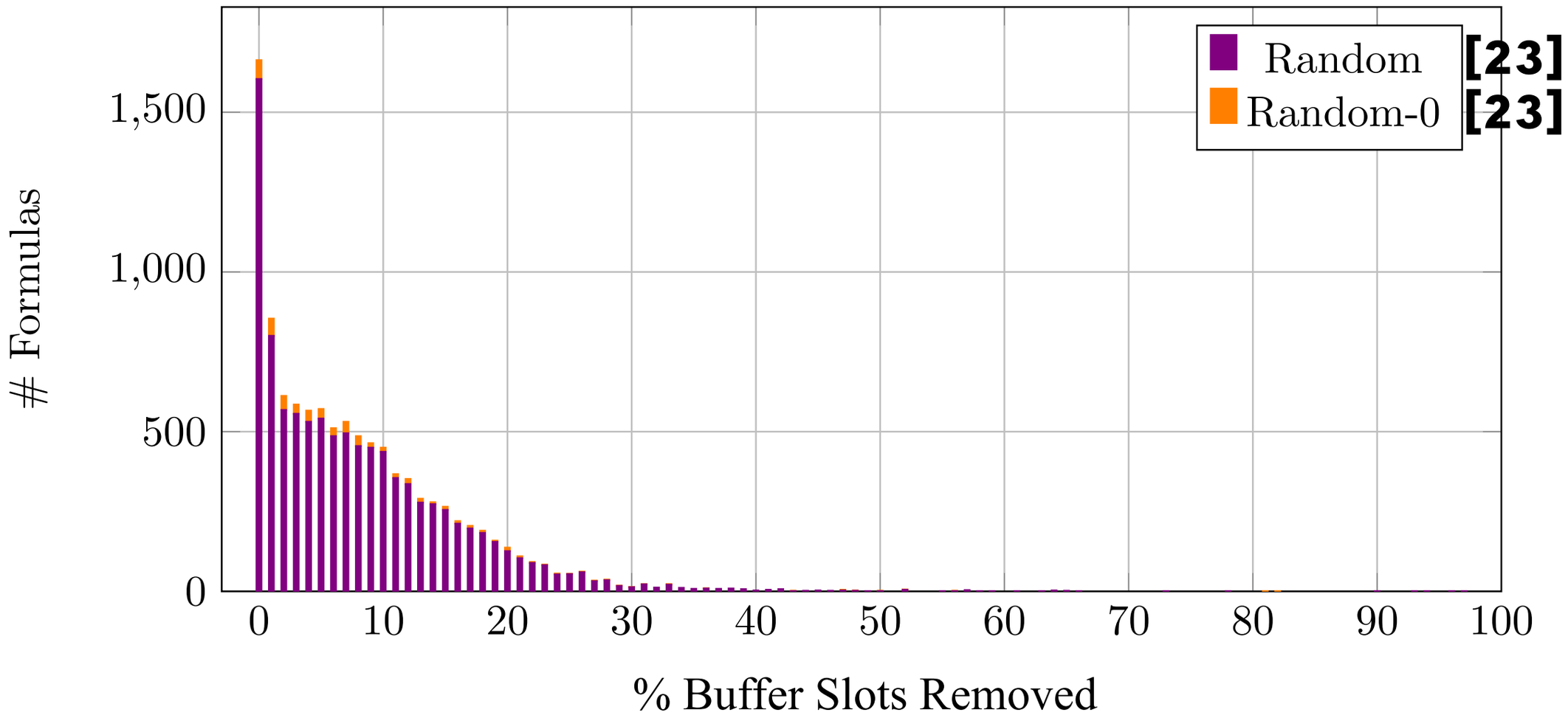
$$\Box_{[0,3]} p \wedge \Box_{[0,5]} q$$
$$\Rightarrow \Box_{[0,3]} (\Box_{[0,0]} p \wedge \Box_{[0,2]} q)$$
$$\Rightarrow \Box_{[0,3]} (p \wedge \Box_{[0,2]} q)$$



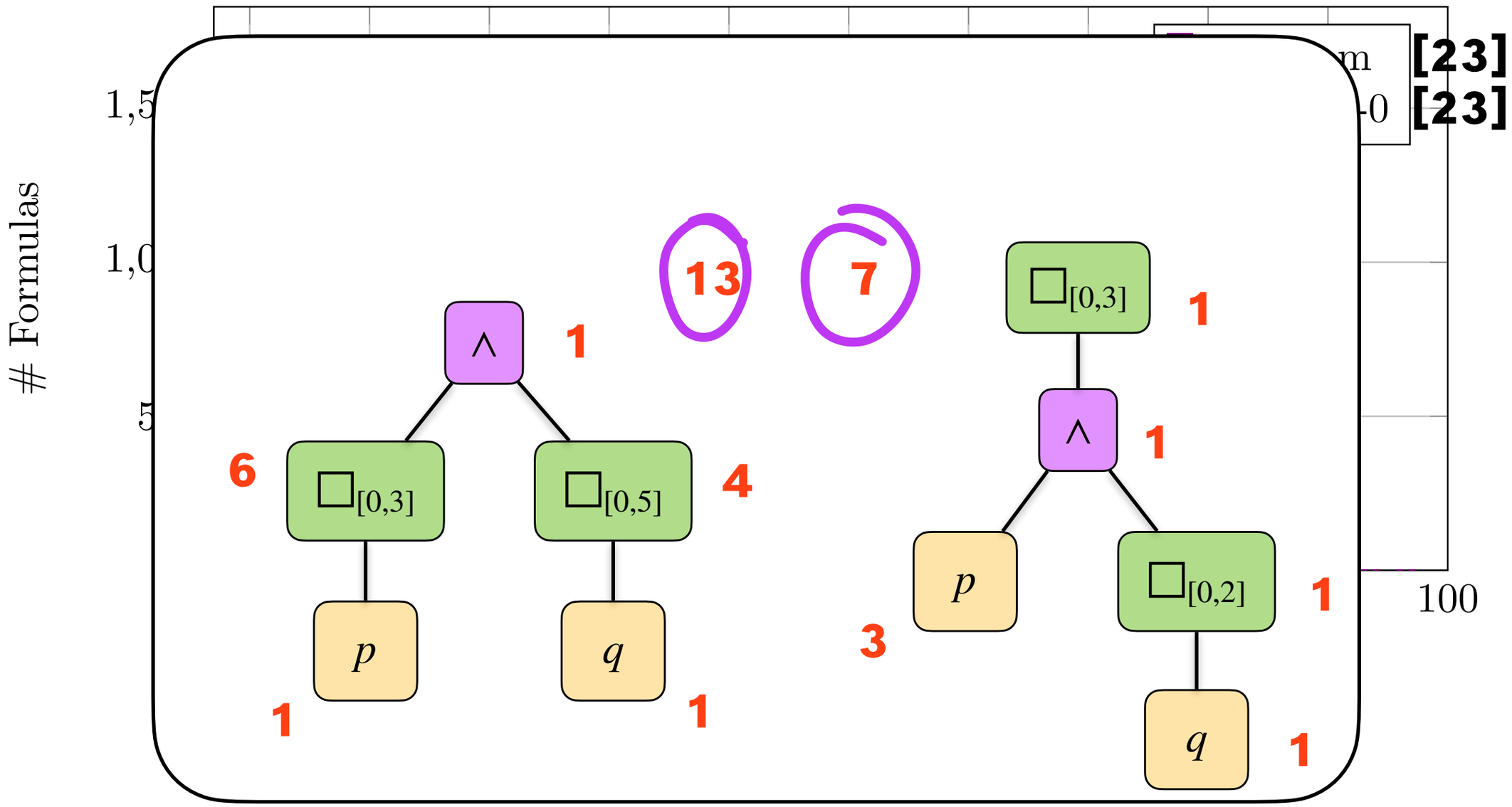
MLTL Equality Saturation Implementation



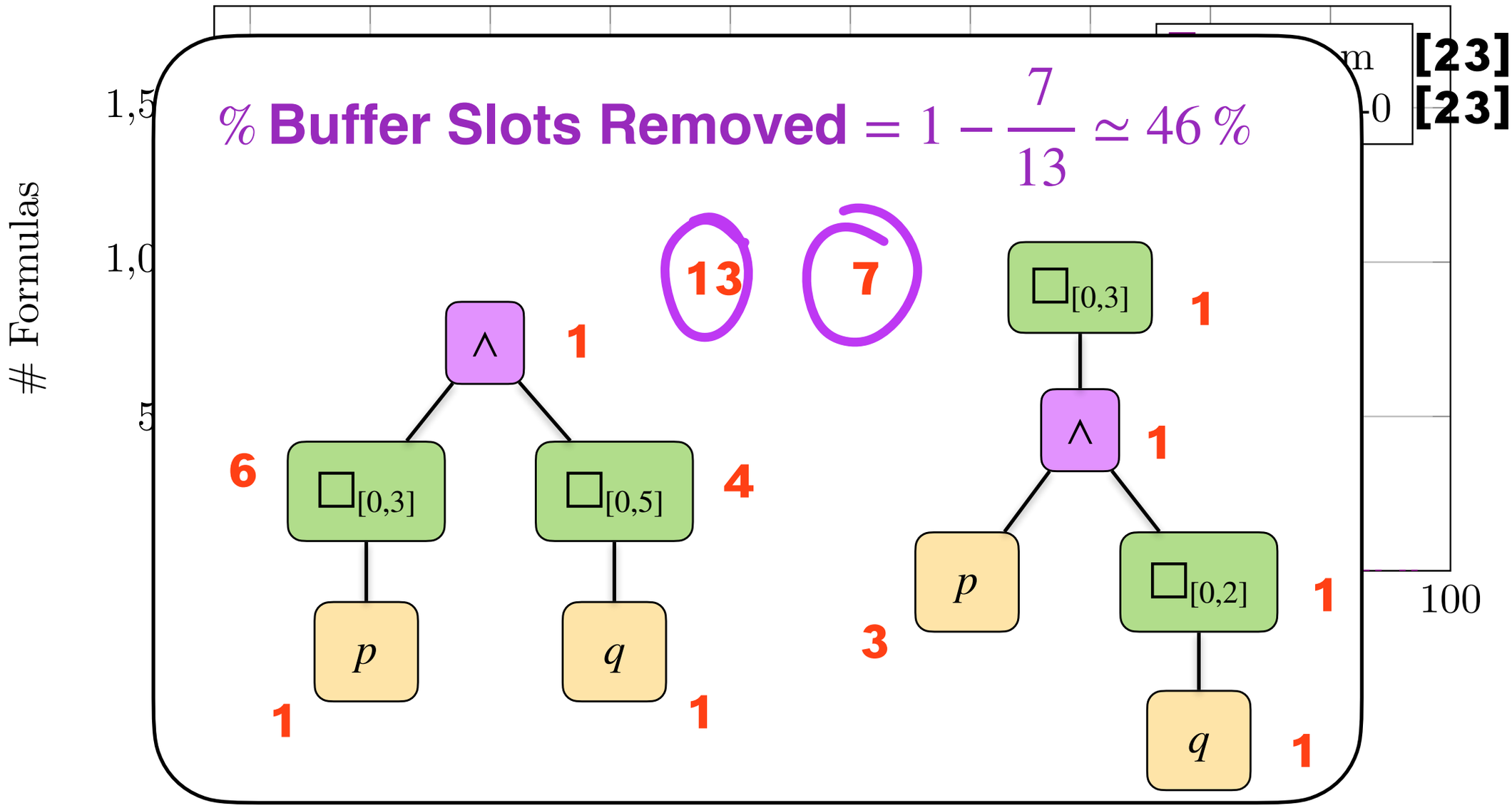
MLTL Equality Saturation Results (Random)



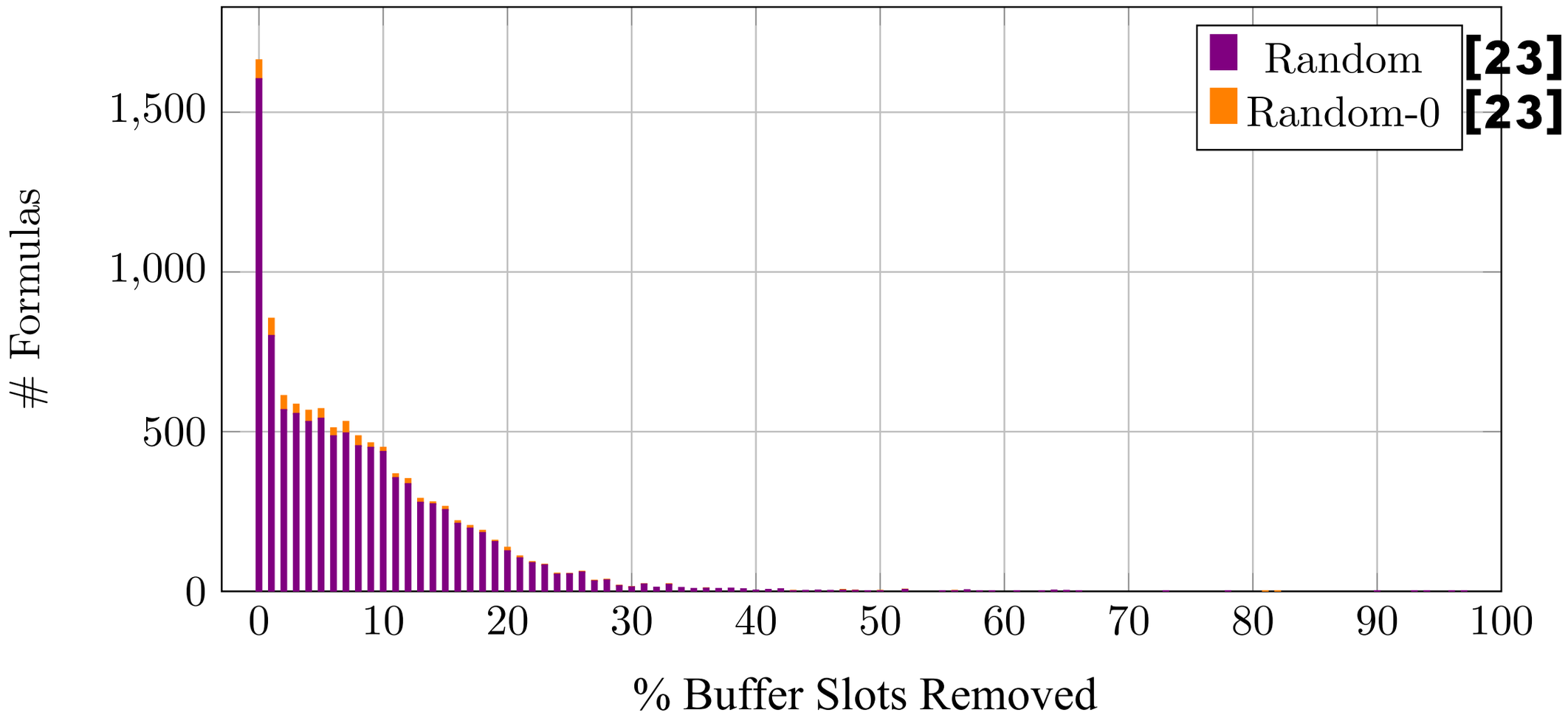
MLTL Equality Saturation Results (Random)



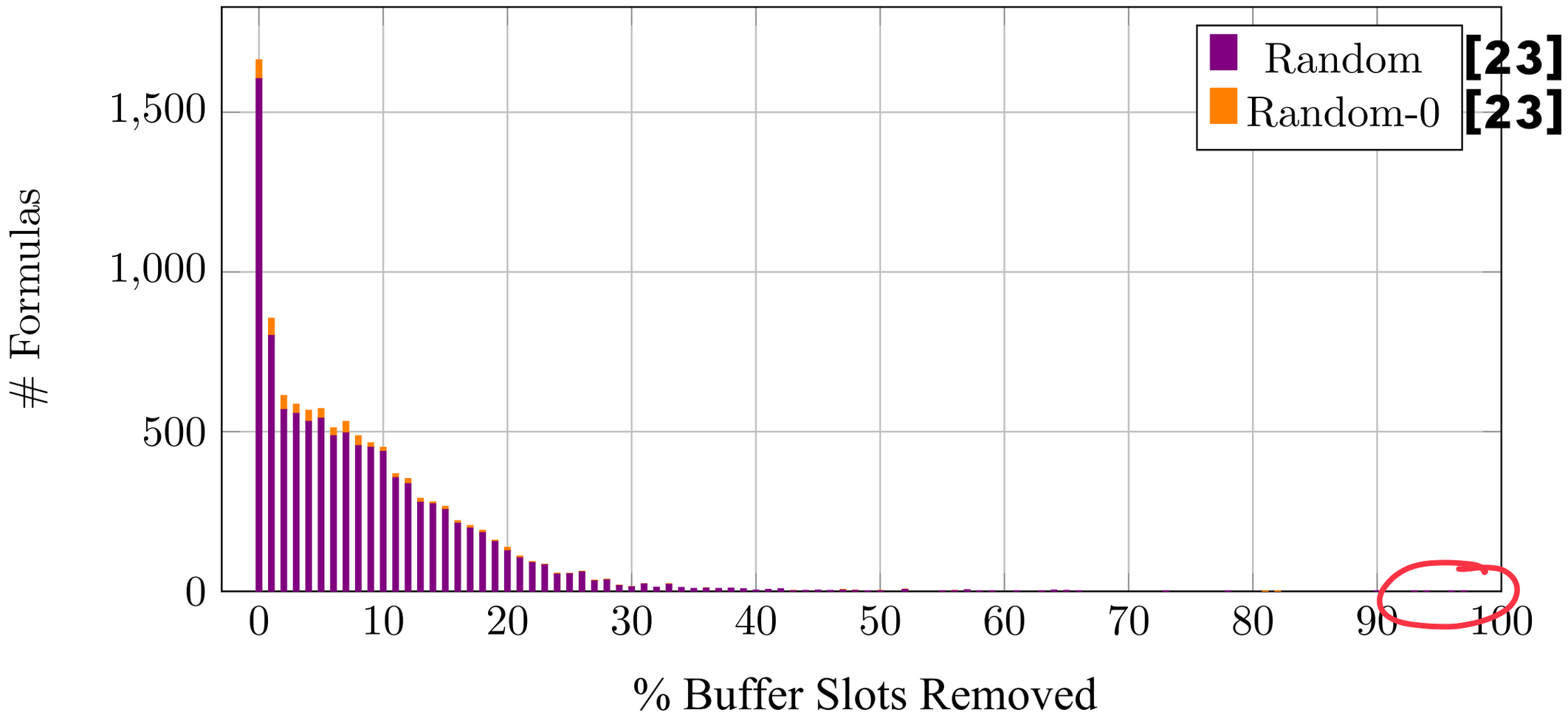
MLTL Equality Saturation Results (Random)



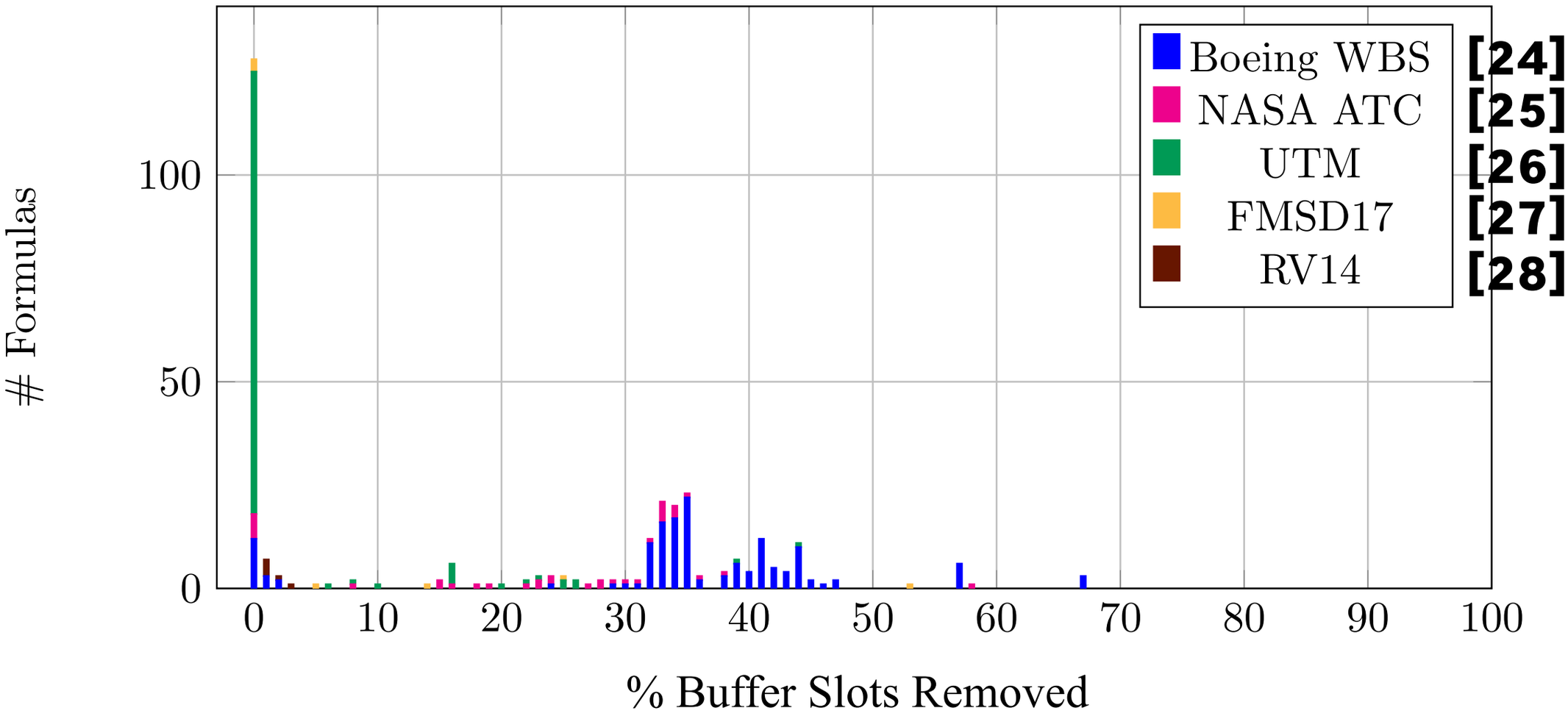
MLTL Equality Saturation Results (Random)



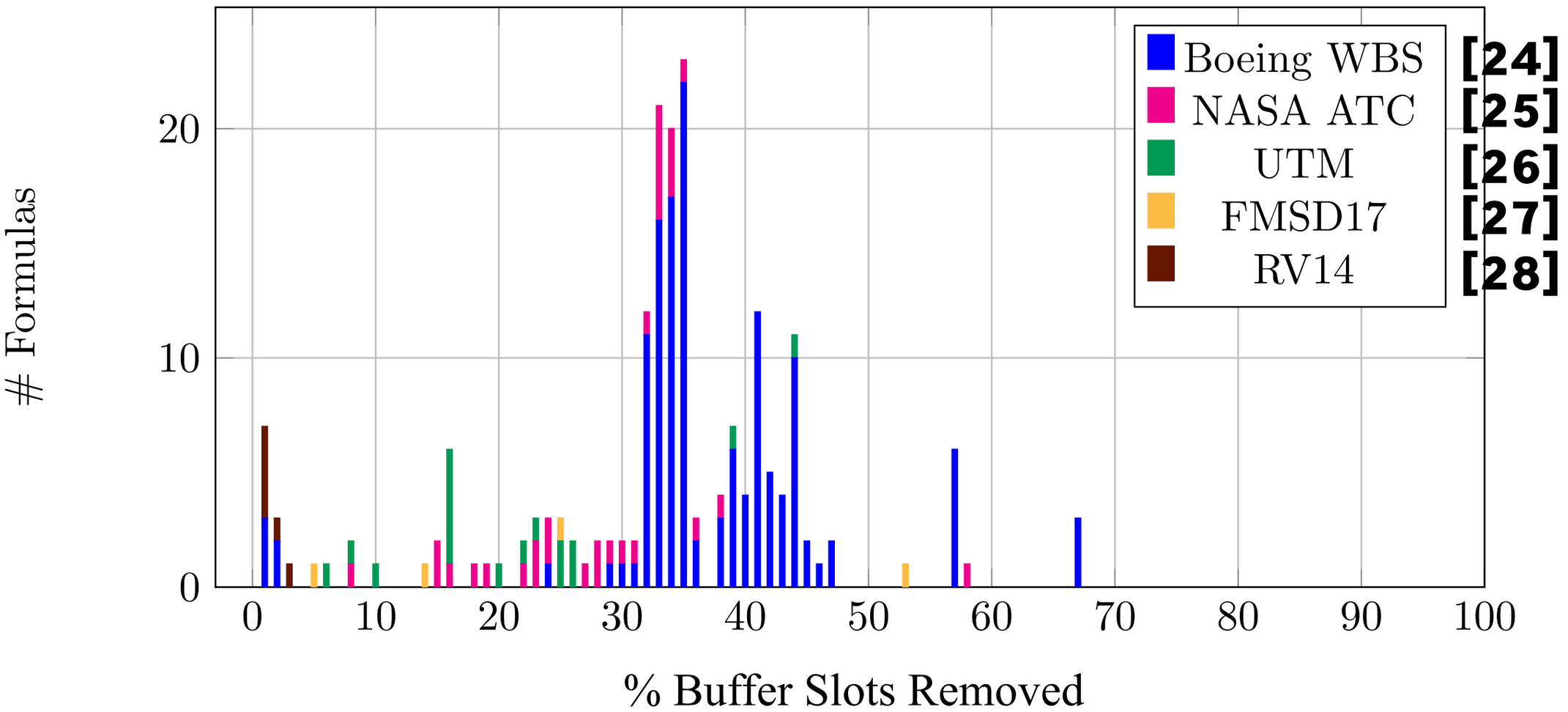
MLTL Equality Saturation Results (Random)



MLTL Equality Saturation Results (Human Authored)



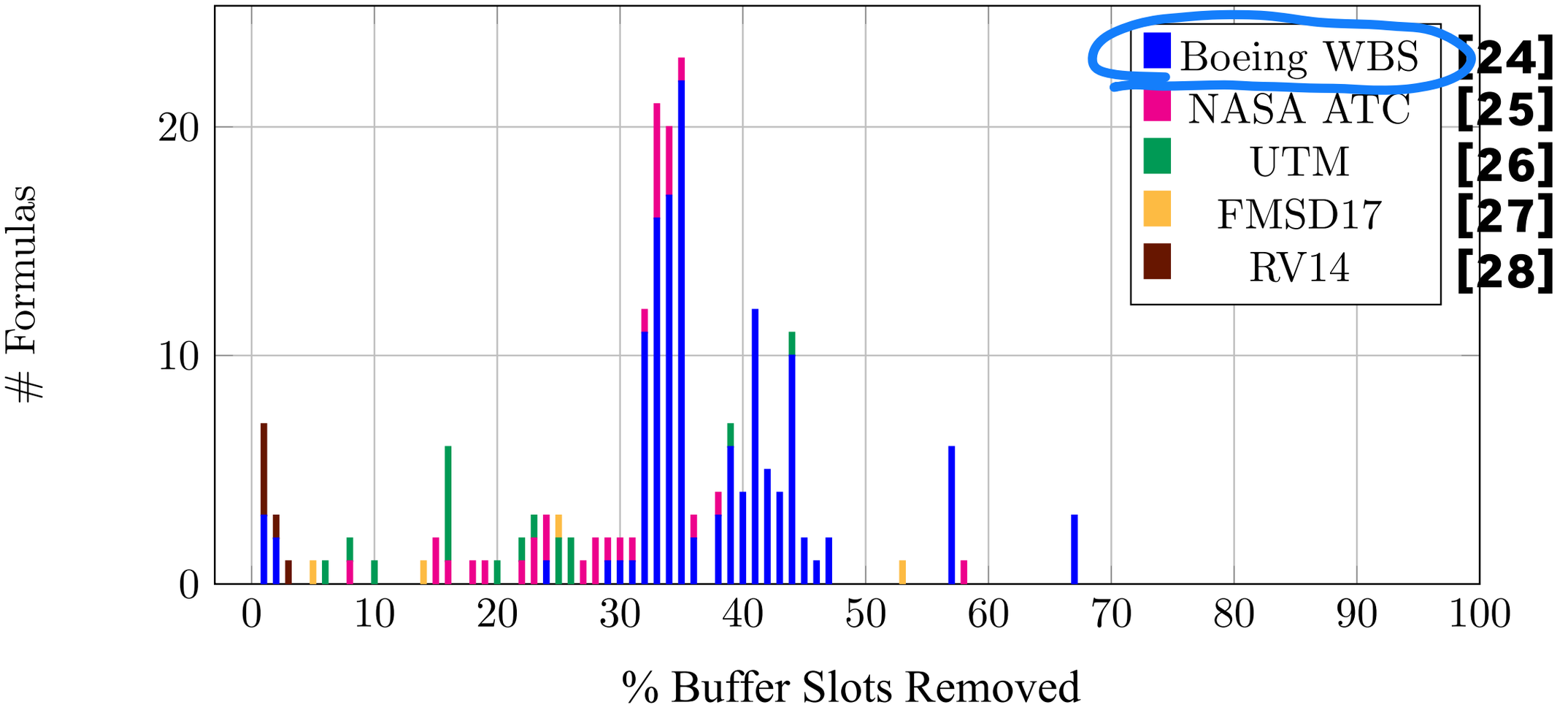
MLTL Equality Saturation Results (Human Authored)



***** < 1% reduction omitted**

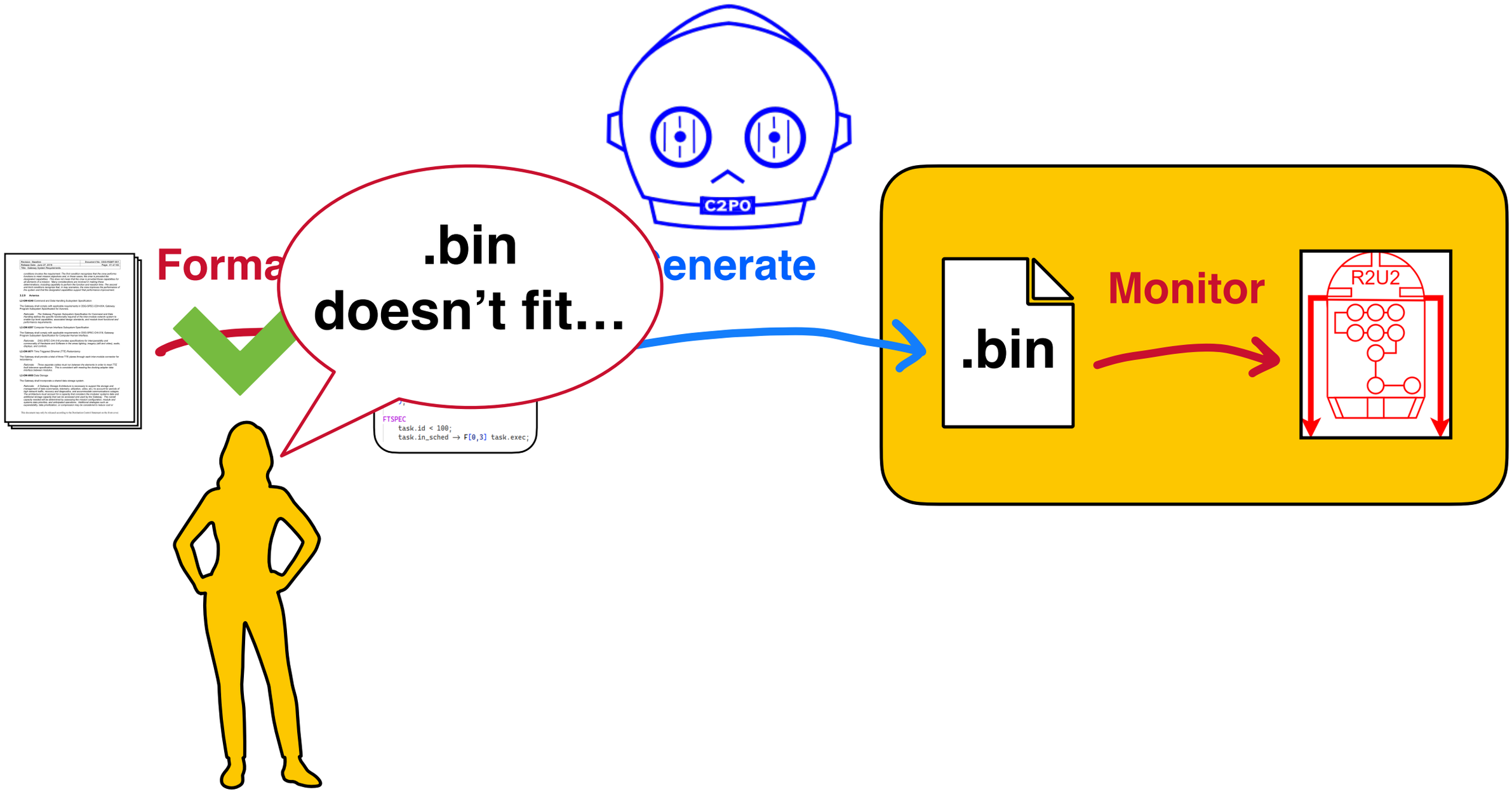
MLTL Equality Saturation Results (Human Authored)

~35% Average Reduction

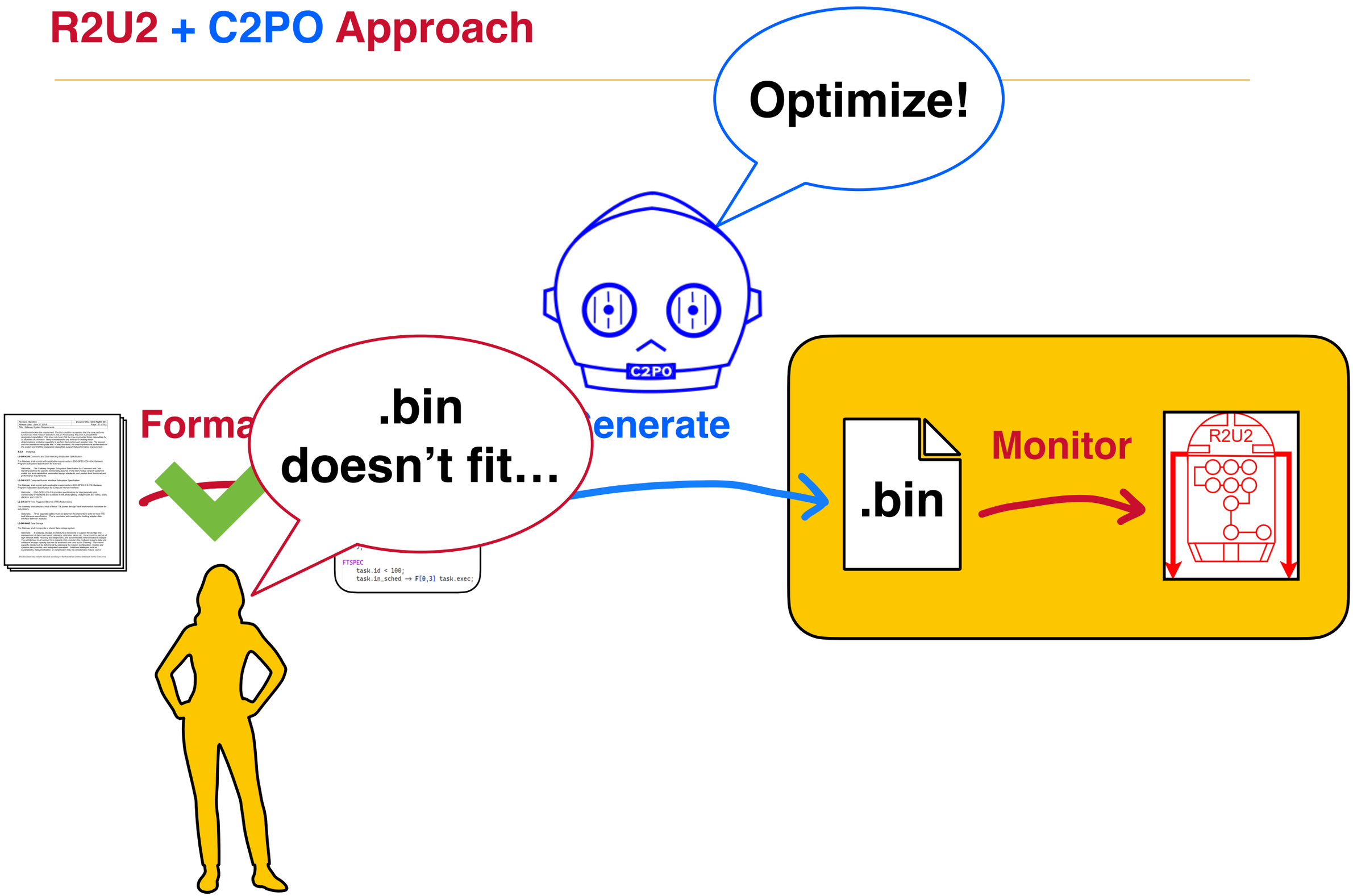


***** < 1% reduction omitted**

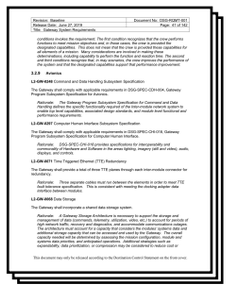
R2U2 + C2PO Approach



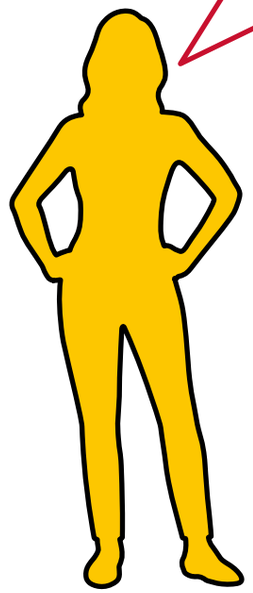
R2U2 + C2PO Approach



R2U2 + C2PO Approach

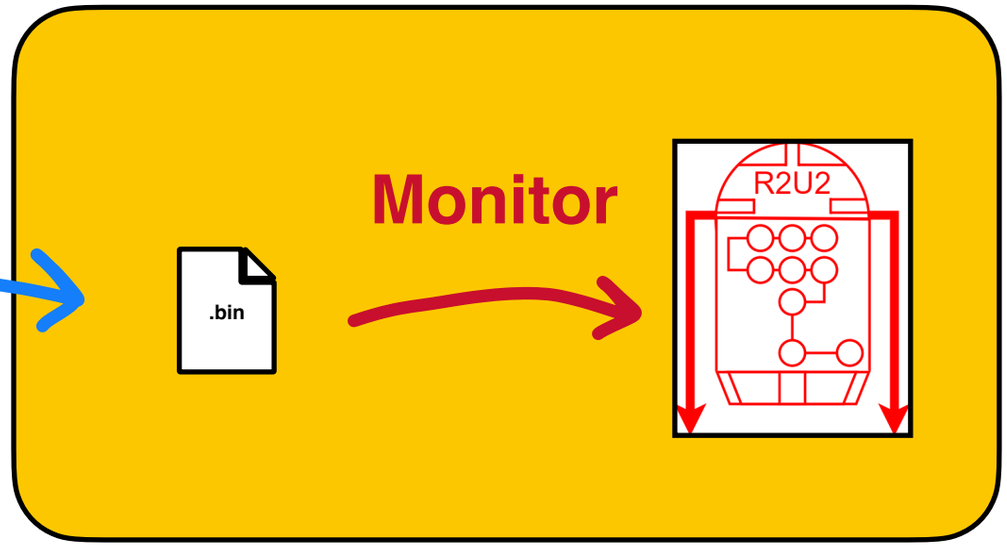


Forma

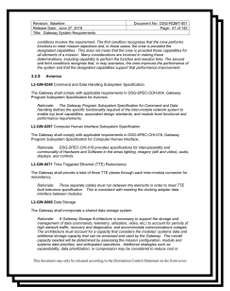


enerate


```
FTSPEC
    task.id < 100;
    task.in_sched → F[0,3] task.exec;
```



R2U2 + C2PO Approach



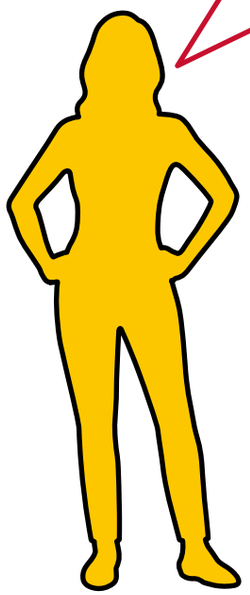
Forma

You're welcome!

Thanks, C2PO!

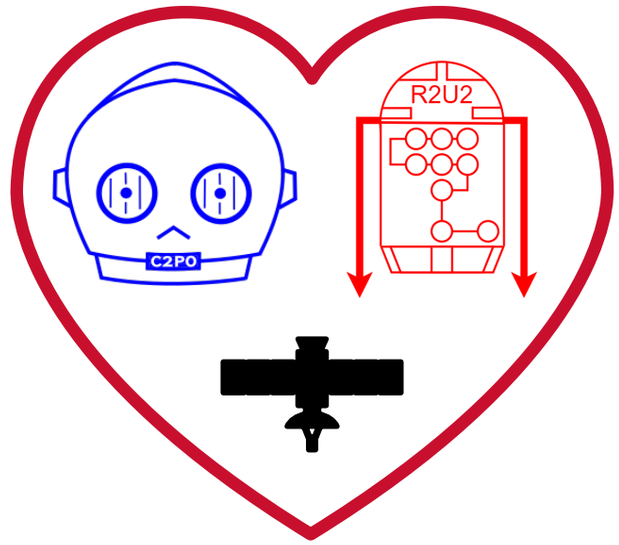
enerate

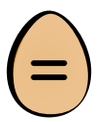
```
FTSPEC
    task.id < 100;
    task.in_sched → F[0,3] task.exec;
```



Questions?

- **C2PO**
  - Optimized encoding of MLTL monitors for R2U2
- FO-MLTL
  - First-order MLTL over bounded dynamic sets
- **MLTL Optimizations**
  - Rewrite Rules
  - Equality saturation via EggLog
- **Future Directions**
  - Implement FO-MLTL in R2U2
  - More rewrites!





r2u2.temporallogic.org

References

[1] World’s First International X-Plane, https://images.nasa.gov/details/EC94-42478-03

[2] Artist concept of a satellite in orbit above earth, https://images.nasa.gov/details/S81-38291

[3] TROPICS Rocket Launch, https://images.nasa.gov/details/KSC-20230530-PH-RKL02_0004

[4] Adamek, Christopher. Gateway system requirements. No. JSC-E-DAA-TN71173. 2019.

[5] d'Angelo, Ben, Sriram Sankaranarayanan, César Sánchez, Will Robinson, Bernd Finkbeiner, Henny B. Sipma, Sandeep Mehrotra, and Zohar Manna. "LOLA: runtime monitoring of synchronous systems." In 12th International Symposium on Temporal Representation and Reasoning (TIME'05), pp. 166-174. IEEE, 2005.

[6] Cimatti, Alessandro, Chun Tian, and Stefano Tonetta. "NuRV: a nuXmv extension for runtime verification." In Runtime Verification: 19th International Conference, RV 2019, Porto, Portugal, October 8–11, 2019, Proceedings 19, pp. 382-392. Springer International Publishing, 2019.

[7] Havelund, Klaus, and Doron Peled. "Efficient runtime verification of first-order temporal properties." In Model Checking Software: 25th International Symposium, SPIN 2018, Malaga, Spain, June 20-22, 2018, Proceedings 25, pp. 26-47. Springer International Publishing, 2018.

[8] Havelund, Klaus. "Rule-based runtime verification revisited." International Journal on Software Tools for Technology Transfer 17 (2015): 143-170.

References

[9] Chen, Feng, and Grigore Roşu. "Mop: an efficient and generic runtime verification framework." In Proceedings of the 22nd annual ACM SIGPLAN conference on Object-oriented programming systems, languages and applications, pp. 569-588. 2007.

[10] Basin, David, Bhargav Nagaraja Bhatt, Srđan Krstić, and Dmitriy Traytel. "Almost event-rate independent monitoring." Formal Methods in System Design 54 (2019): 449-478.

[11] Goldberg, Allen, and Klaus Havelund. "Automated Runtime Verification with Eagle." In MSVVEIS. 2005.

[12] Barringer, Howard, Klaus Havelund, David Rydeheard, and Alex Groce. "Rule systems for runtime verification: A short tutorial." In Runtime Verification: 9th International Workshop, RV 2009, Grenoble, France, June 26-28, 2009. Selected Papers 9, pp. 1-24. Springer Berlin Heidelberg, 2009.

[13] Hallé, Sylvain, Tevfik Bultan, Graham Hughes, Muath Alkhalaf, and Roger Villemaire. "Runtime Verification of Web Service Interface Contracts." Computer 43, no. 3 (2010): 59-66.

[14] Schneider, Joshua, David Basin, Srđan Krstić, and Dmitriy Traytel. "A formally verified monitor for metric first-order temporal logic." In Runtime Verification: 19th International Conference, RV 2019, Porto, Portugal, October 8–11, 2019, Proceedings 19, pp. 310-328. Springer International Publishing, 2019.

References

[15] Basin, David A., Felix Klaedtke, and Eugen Zalinescu. "The MonPoly Monitoring Tool." RV-CuBES 3 (2017): 19-28.

[16] Reinbacher, Thomas, Kristin Yvonne Rozier, and Johann Schumann. "Temporal-logic based runtime observer pairs for system health management of real-time systems." In Tools and Algorithms for the Construction and Analysis of Systems: 20th International Conference, TACAS 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014. Proceedings 20, pp. 357-372. Springer Berlin Heidelberg, 2014.

[17] Johannsen, Chris, Phillip Jones, Brian Kempa, Kristin Yvonne Rozier, and Pei Zhang. "R2U2 Version 3.0: Re-Imagining a Toolchain for Specification, Resource Estimation, and Optimized Observer Generation for Runtime Verification in Hardware and Software." In International Conference on Computer Aided Verification, pp. 483-497. Cham: Springer Nature Switzerland, 2023.

[18] Kempa, Brian, Pei Zhang, Phillip H. Jones, Joseph Zambreno, and Kristin Yvonne Rozier. "Embedding online runtime verification for fault disambiguation on Robonaut2." In International Conference on Formal Modeling and Analysis of Timed Systems, pp. 196-214. Cham: Springer International Publishing, 2020.

References

[19] Johannsen, Chris, Brian Kempa, Phillip H. Jones, Kristin Y. Rozier, and Tichakorn Wongpiromsarn. "Impossible Made Possible: Encoding Intractable Specifications via Implied Domain Constraints." In International Conference on Formal Methods for Industrial Critical Systems, pp. 151-169. Cham: Springer Nature Switzerland, 2023.

[20] Dershowitz, Nachum, and Jean-Pierre Jouannaud. "Rewrite systems." In Formal models and semantics, pp. 243-320. Elsevier, 1990.

[21] Tate, Ross, Michael Stepp, Zachary Tatlock, and Sorin Lerner. "Equality saturation: a new approach to optimization." In Proceedings of the 36th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages, pp. 264-276. 2009.

[22] Zhang, Yihong, Yisu Remy Wang, Oliver Flatt, David Cao, Philip Zucker, Eli Rosenthal, Zachary Tatlock, and Max Willsey. "Better together: Unifying datalog and equality saturation." Proceedings of the ACM on Programming Languages 7, no. PLDI (2023): 468-492.

[23] Li, Jianwen, Moshe Y. Vardi, and Kristin Y. Rozier. "Satisfiability checking for mission-time LTL." In Computer Aided Verification: 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part II 31, pp. 3-22. Springer International Publishing, 2019.

References

[24] Bozzano, Marco, Alessandro Cimatti, Anthony Fernandes Pires, David Jones, Greg Kimberly, T. Petri, R. Robinson, and Stefano Tonetta. "Formal design and safety analysis of AIR6110 wheel brake system." In Computer Aided Verification: 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part I 27, pp. 518-535. Springer International Publishing, 2015.

[25] Gario, Marco, Alessandro Cimatti, Cristian Mattarei, Stefano Tonetta, and Kristin Yvonne Rozier. "Model checking at scale: Automated air traffic control design space exploration." In Computer Aided Verification: 28th International Conference, CAV 2016, Toronto, ON, Canada, July 17-23, 2016, Proceedings, Part II 28, pp. 3-22. Springer International Publishing, 2016.

[26] Cauwels, Matthew, Abigail Hammer, Benjamin Hertz, Phillip H. Jones, and Kristin Y. Rozier. "Integrating runtime verification into an automated uas traffic management system." In European Conference on Software Architecture, pp. 340-357. Cham: Springer International Publishing, 2020.

[27] Moosbrugger, P., Rozier, K.Y. & Schumann, J. R2U2: monitoring and diagnosis of security threats for unmanned aerial systems. Form Methods Syst Des 51, 31–61 (2017). https://doi.org/10.1007/s10703-017-0275-x

[28] Geist, Johannes, Kristin Y. Rozier, and Johann Schumann. "Runtime observer pairs and Bayesian network reasoners on-board FPGAs: flight-certifiable system health management for embedded systems." In International Conference on Runtime Verification, pp. 215-230. Cham: Springer International Publishing, 2014.

References

[29] De Moura, Leonardo, and Nikolaj Bjørner. "Z3: An efficient SMT solver." In International conference on Tools and Algorithms for the Construction and Analysis of Systems, pp. 337-340. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.