Guided GUI Testing of Android Apps with Minimal Restart and Approximate Learning

Wontae Choi, George Necula, and Koushik Sen

University of California, Berkeley

OOPSLA 2013 @ Indianapolis, USA

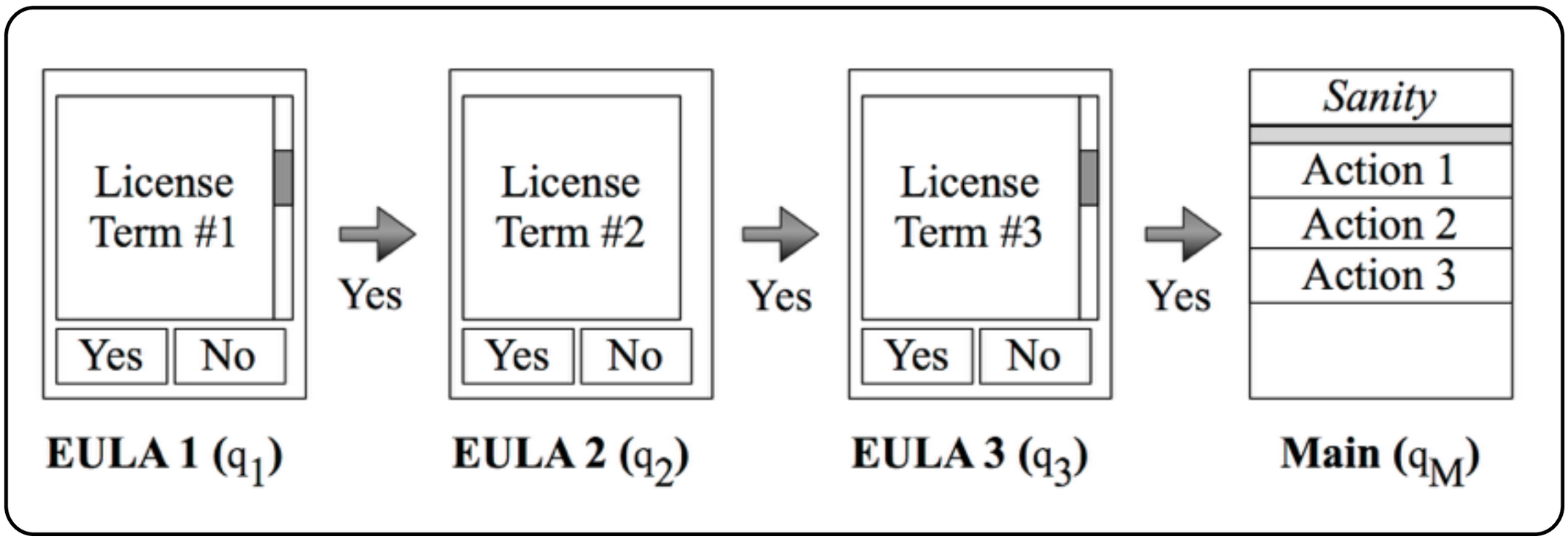
GUI Testing

Modern platforms use rich GUI

This talk : automated GUI testing

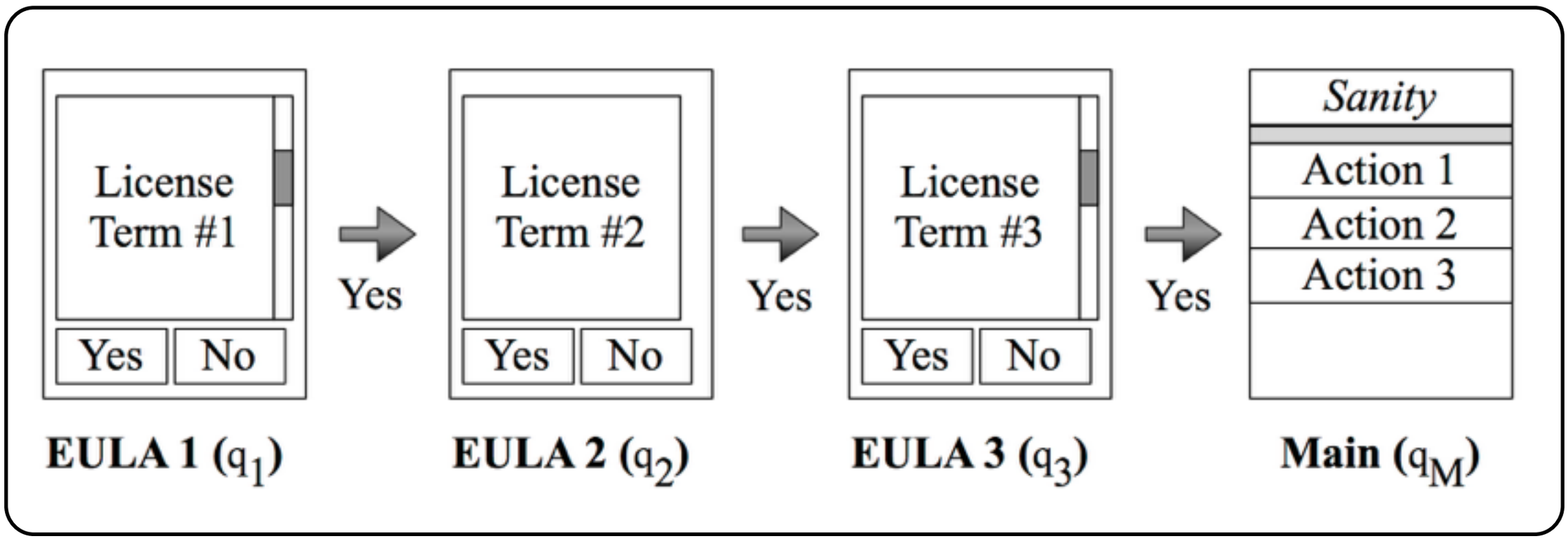
Working Example

Working Example



- Android app
- 3 consecutive EULA screens
- Touching **No** will terminate the app
- **Scroll** does nothing

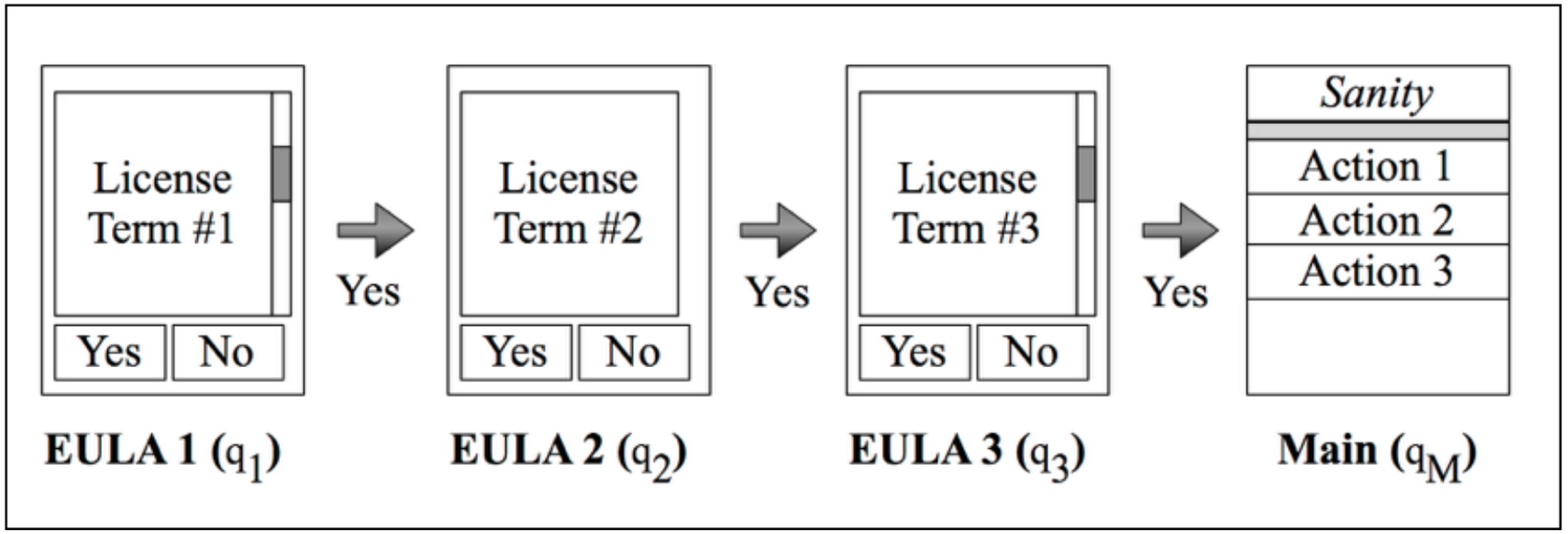
Working Example



- Android app
- 3 consecutive EULA screens
- Touching **No** will terminate the app
- **Scroll** does nothing

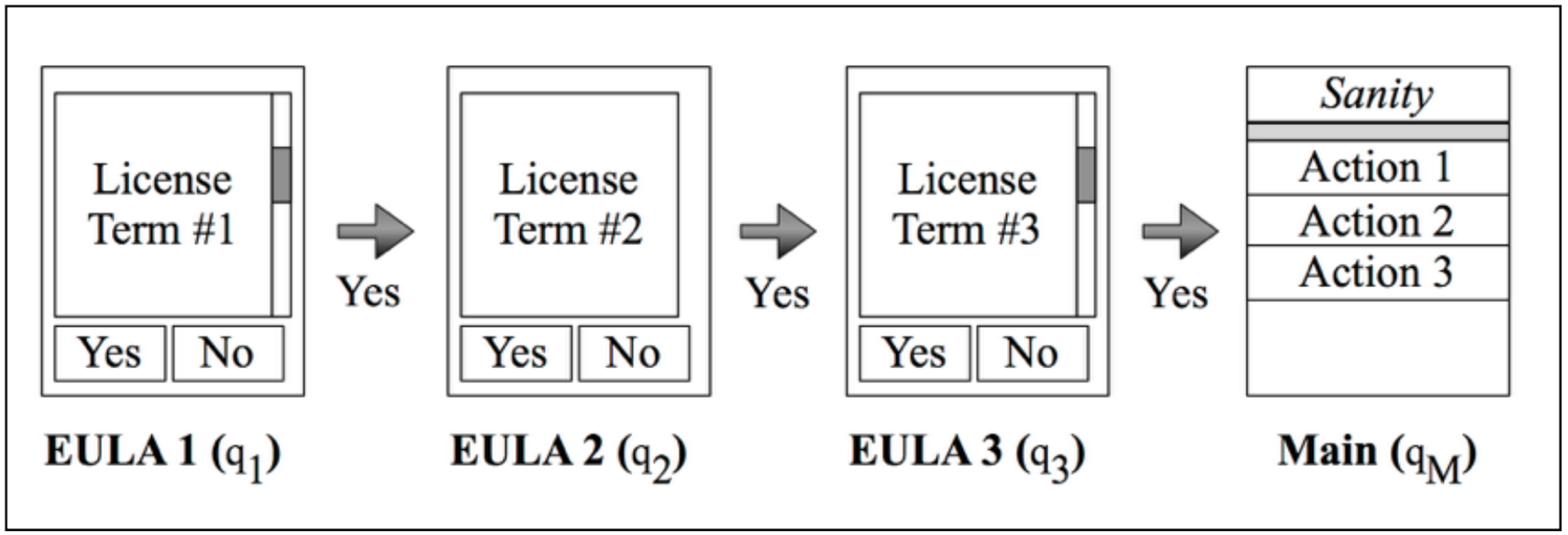
Random Testing

(Existing Idea)



Random Testing

(Existing Idea)

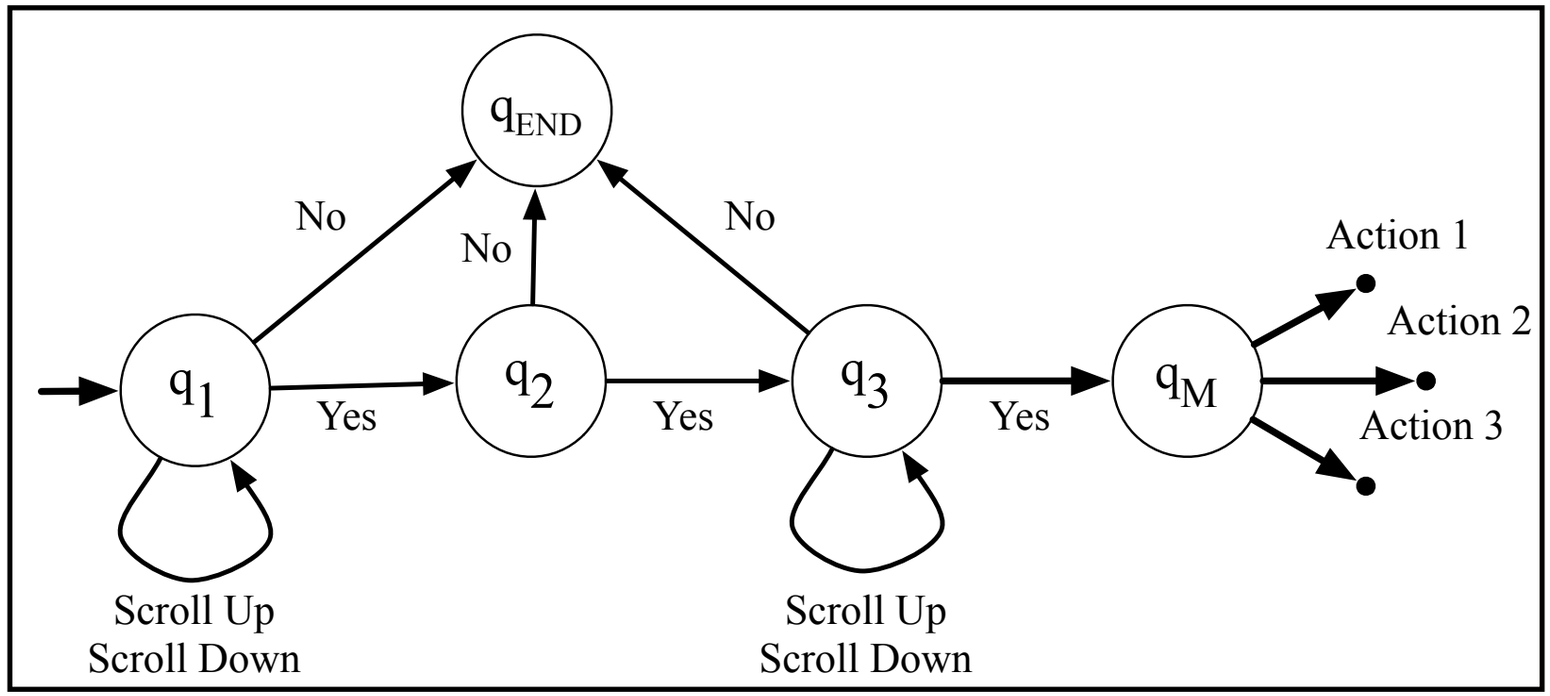


- Randomly tries the enabled events
- Good to find shallow bugs

- May not reach deep program states
  - Difficult to reach the main screen
    - Takes 24 events + 7 restarts (on average)
    - Optimum : 3 events

Model Based Testing

(Existing Idea)



Behavioral model of the example

- A behavioral model is provided
- Computes test cases from the model
  - **[Scroll-Down, Scroll-Up, Yes, Yes, Scroll-Down, Scroll-Up, Yes]** covers all states and non-terminating transitions

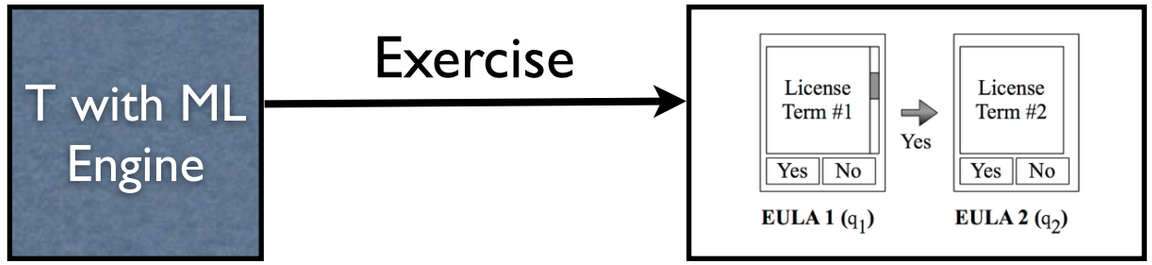
- Manually providing a model is tedious and error-prone

Testing with Model Learning

(Existing Idea)

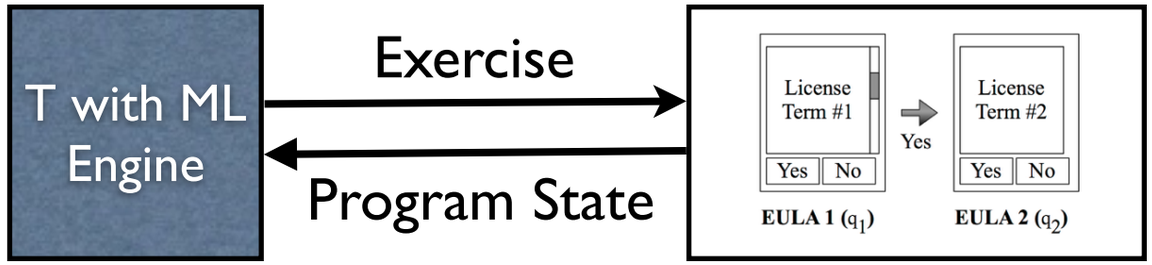
Testing with Model Learning

(Existing Idea)



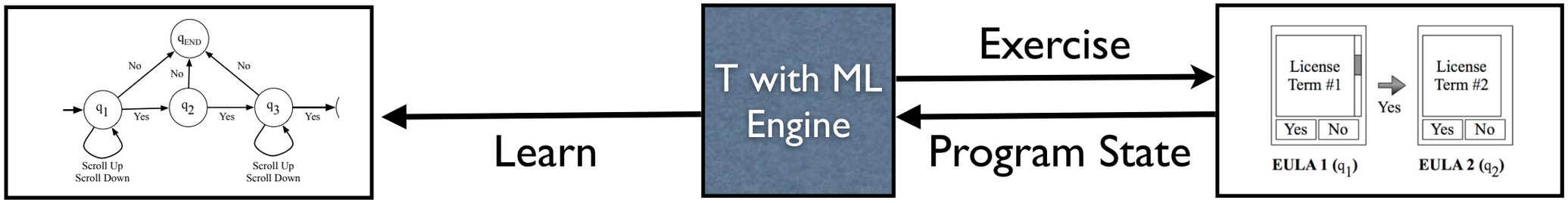
Testing with Model Learning

(Existing Idea)



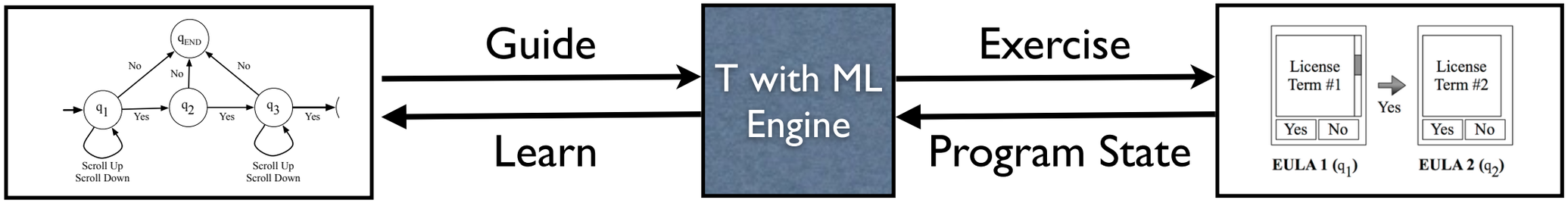
Testing with Model Learning

(Existing Idea)



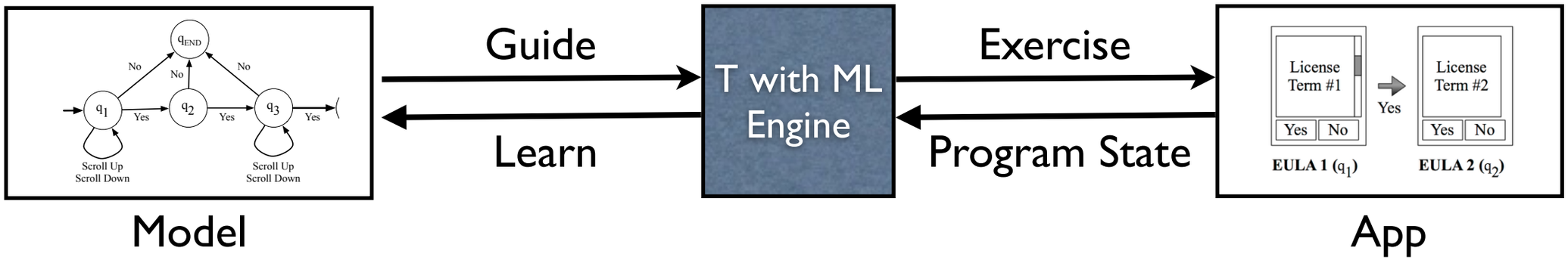
Testing with Model Learning

(Existing Idea)



Testing with Model Learning

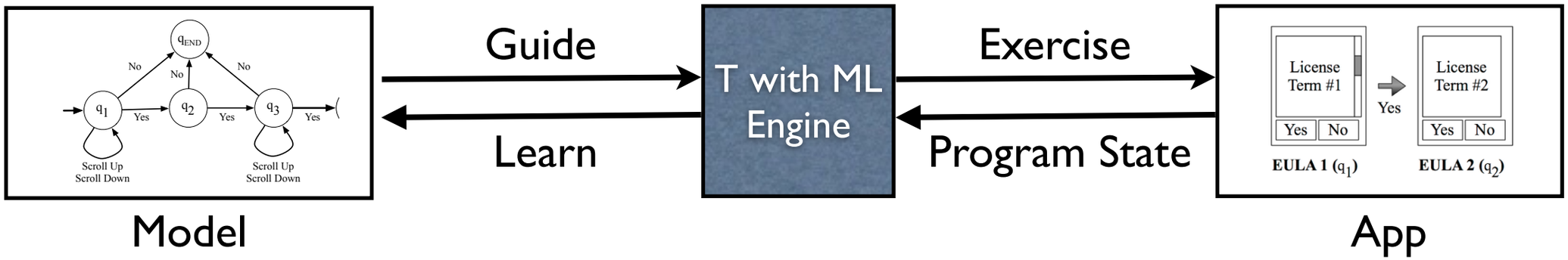
(Existing Idea)



Learn a model during testing
Guide testing using a model

Testing with Model Learning

(Existing Idea)



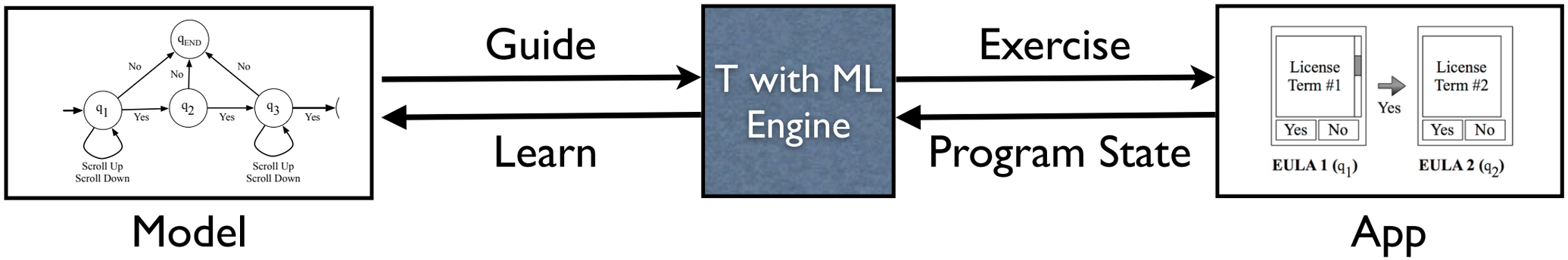
Learn a model during testing
Guide testing using a model

*L** experience

- *L** is a widely-used learning algorithm
- *L** based testing is less effective than random testing
  - *Conservative learning : large model*
  - *Frequently restarts a target application*

Testing with Model Learning

(Existing Idea)



Learn a model during testing
Guide testing using a model

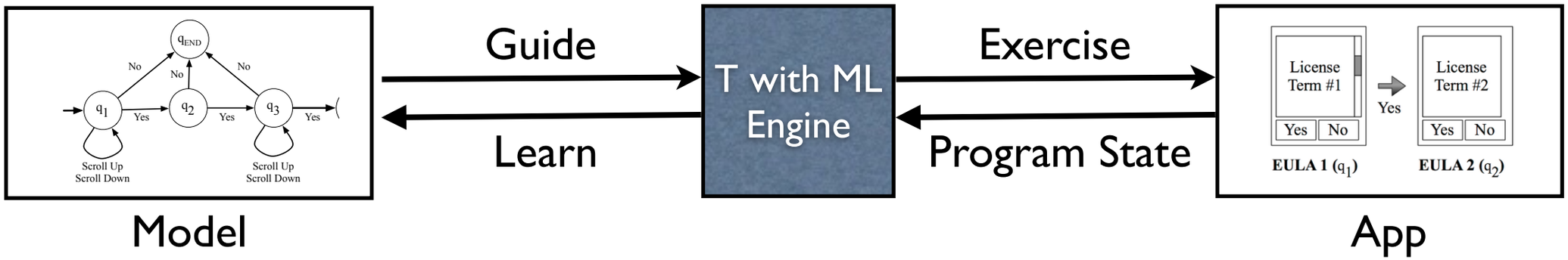
*L** experience

- *L** is a widely-used learning algorithm
- *L** based testing is less effective tha
  - *Conservative learning : large model*
  - *Frequently restarts a target application*

with similar states !

Testing with Model Learning

(Existing Idea)



Learn a model during testing
Guide testing using a model

*L** experience

- *L** is a widely-used learning algorithm
- *less effective tha*
  - *Conservative learning : large model*
  - *Frequently restarts a target application*

execution cost
1 restart = 5 GUI events

with similar states !

SwiftHand Algorithm

Key Insight

SwiftHand Algorithm

Key Insight

- Explore diverse program states quickly
  - Optimistically keep the model small
  - Refine the model when encountering inconsistency

SwiftHand Algorithm

Key Insight

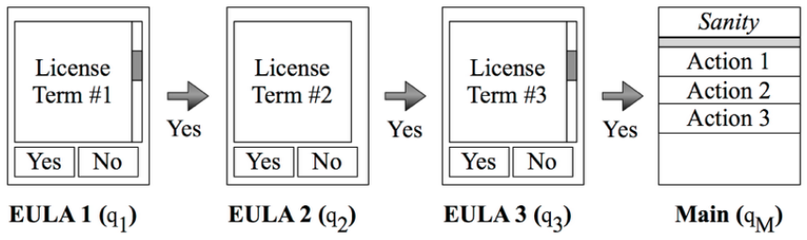
- Explore diverse program states quickly
  - Optimistically keep the model small
  - Refine the model when encountering inconsistency

- Be aware of the cost model for different events
  - Try to reach a goal state without restart

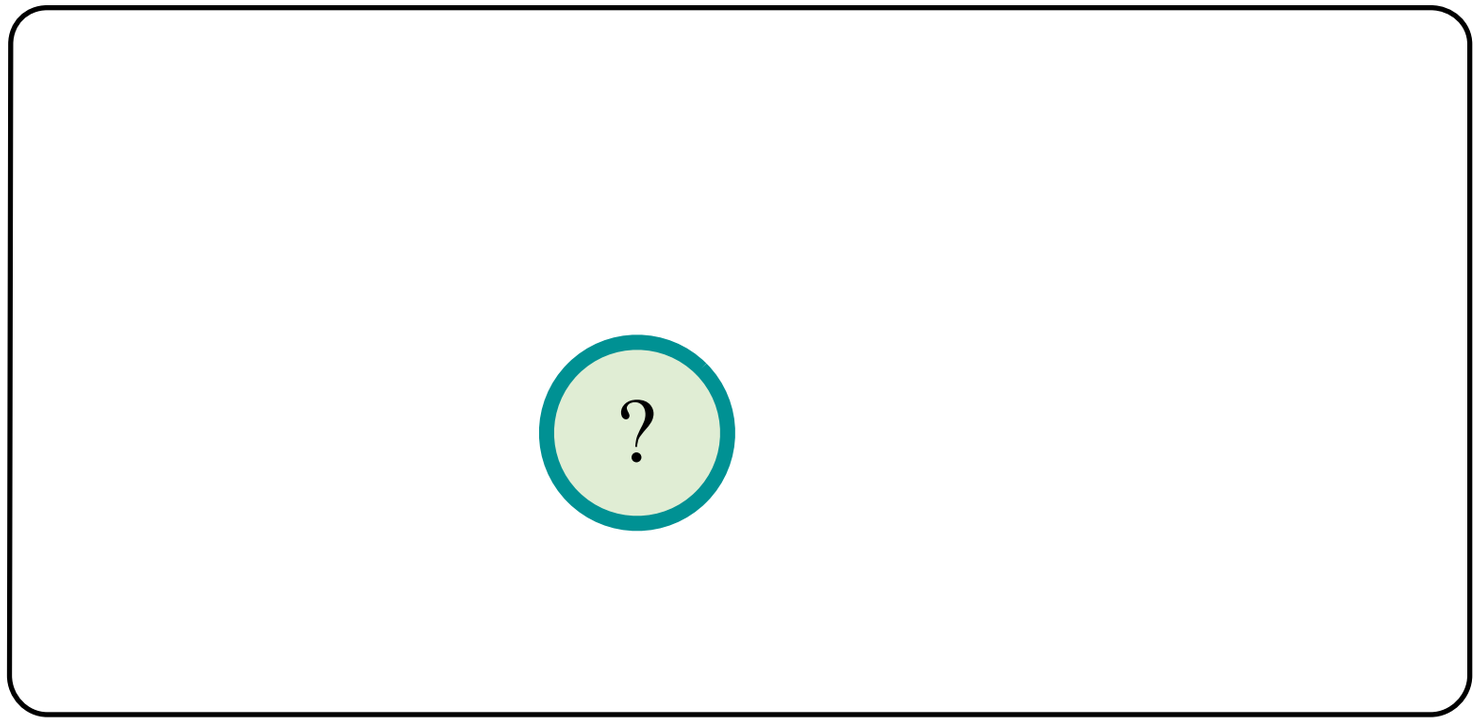
SwiftHand with Example

Initialization

Actual Application



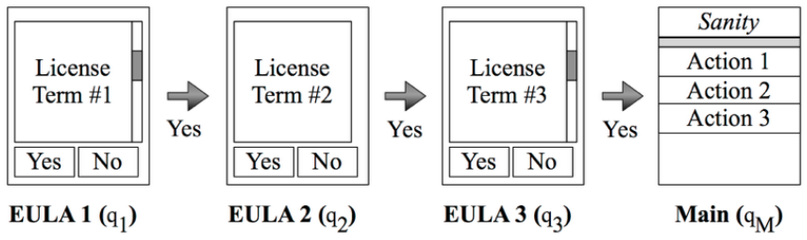
Model Learned



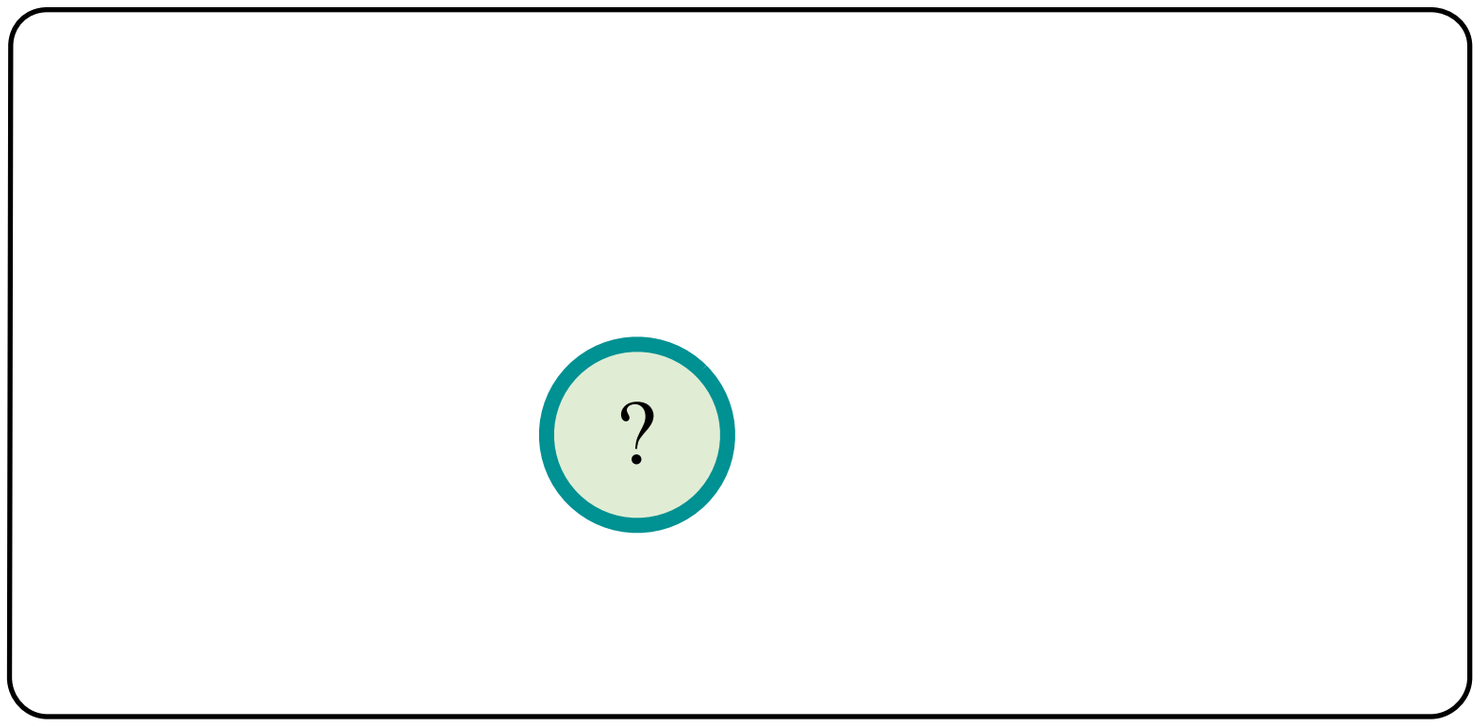
SwiftHand with Example

Initialization

Actual Application



Model Learned

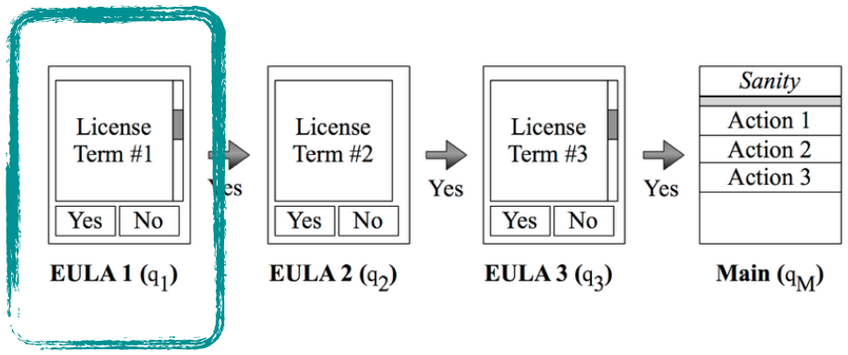


SwiftHand constructs a model
via observing the application screen.

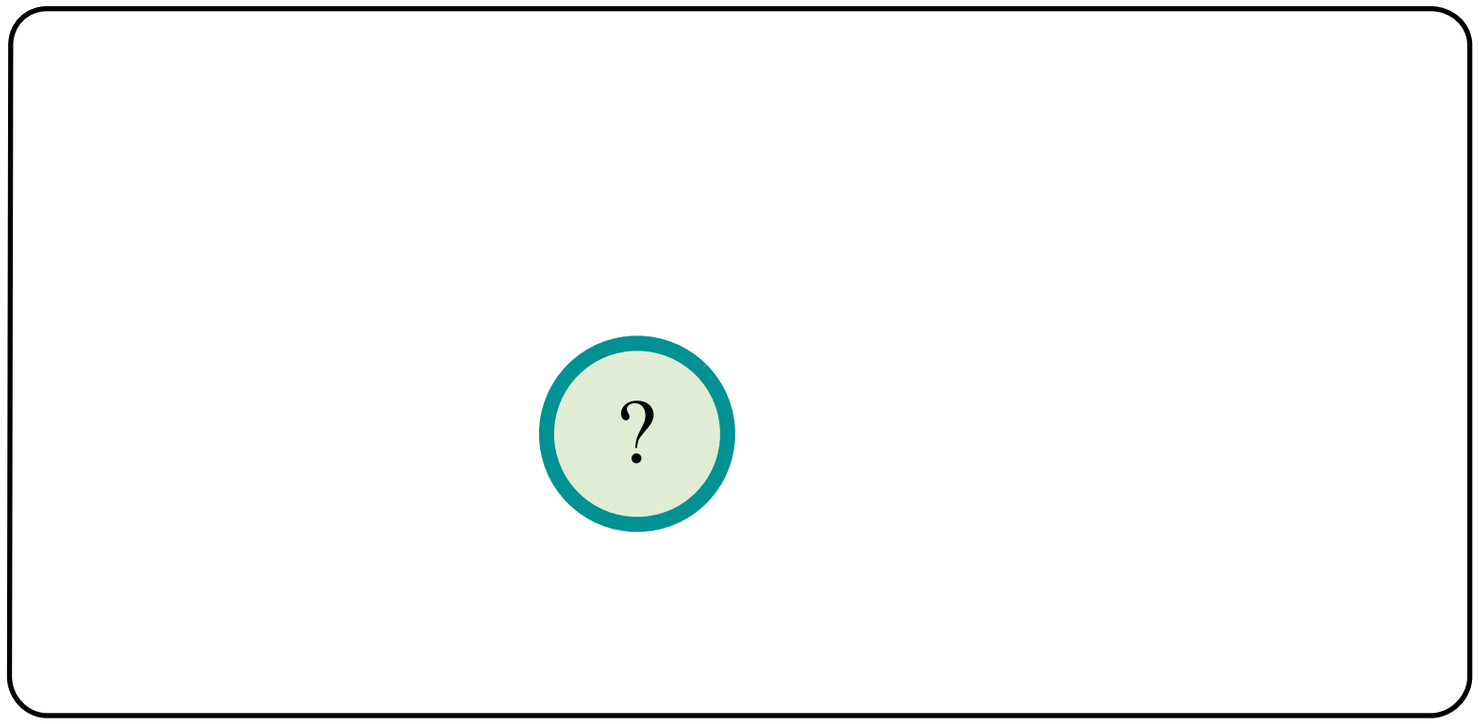
SwiftHand with Example

Initialization

Actual Application



Model Learned

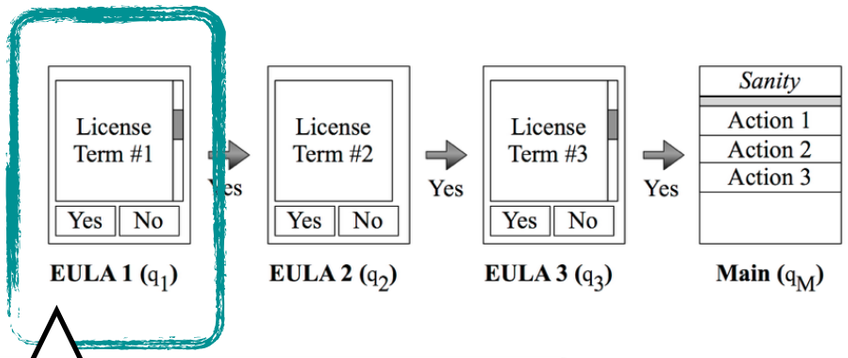


SwiftHand constructs a model
via observing the application screen.

SwiftHand with Example

Initialization

Actual Application



DecorView(1)
+ *LinearLayout*(2)
| + *ScrollableLayout*(3)
| | + *TextBox*(4)
| | *LinearLayout*(5)
| | + *Button*[Yes](6)
| | | *Button*[No](7)

get GUI tree

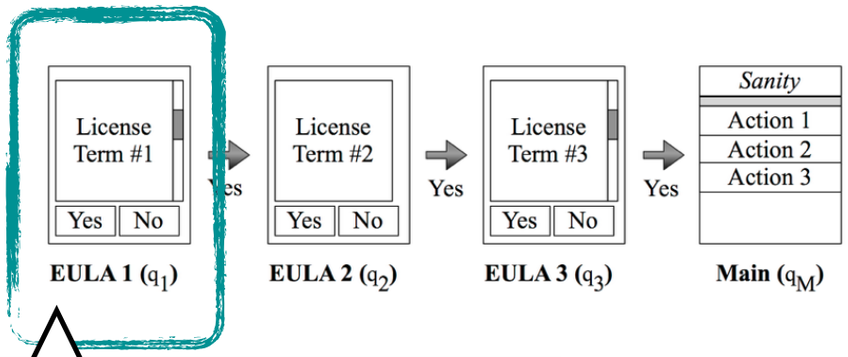
Model Learned

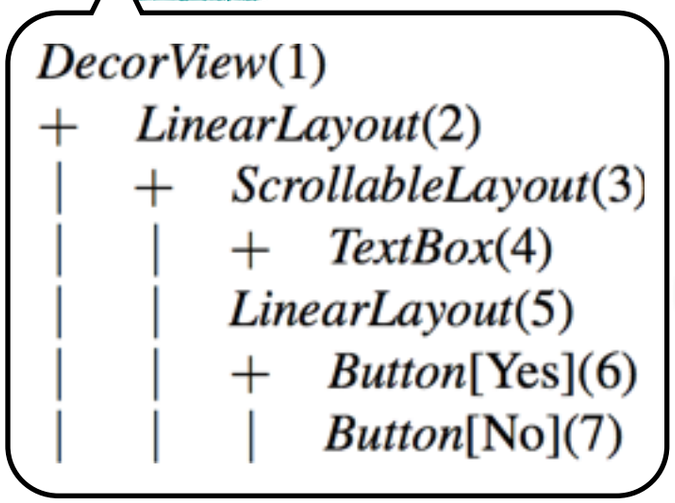
SwiftHand constructs a model
via observing the application screen.

SwiftHand with Example

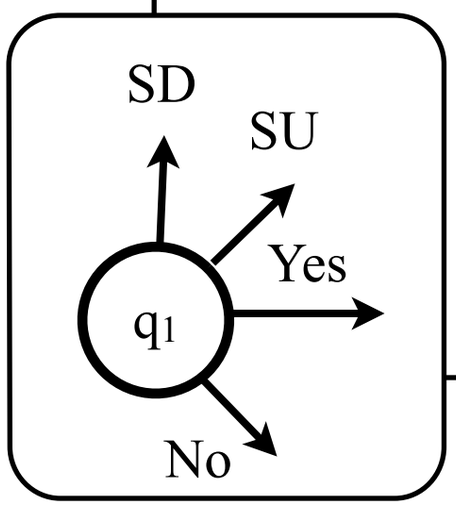
Initialization

Actual Application





get GUI tree



abstract

Model Learned

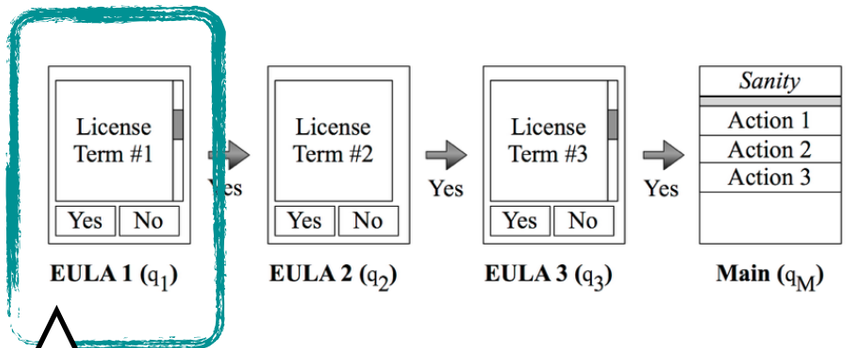
?

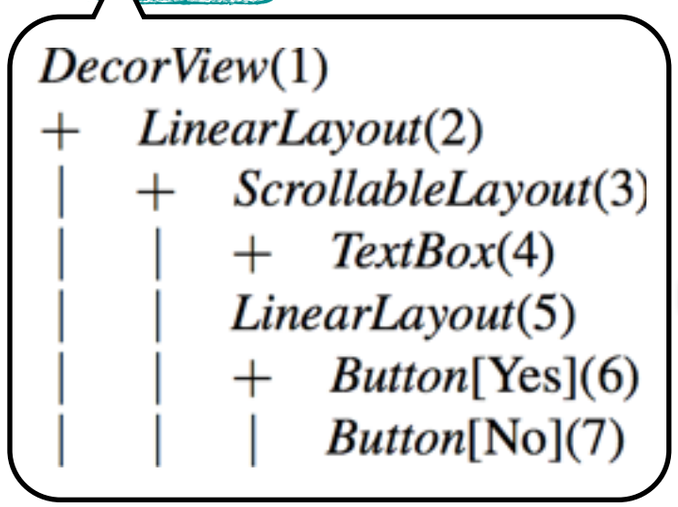
SwiftHand constructs a model
via observing the application screen.

SwiftHand with Example

Initialization

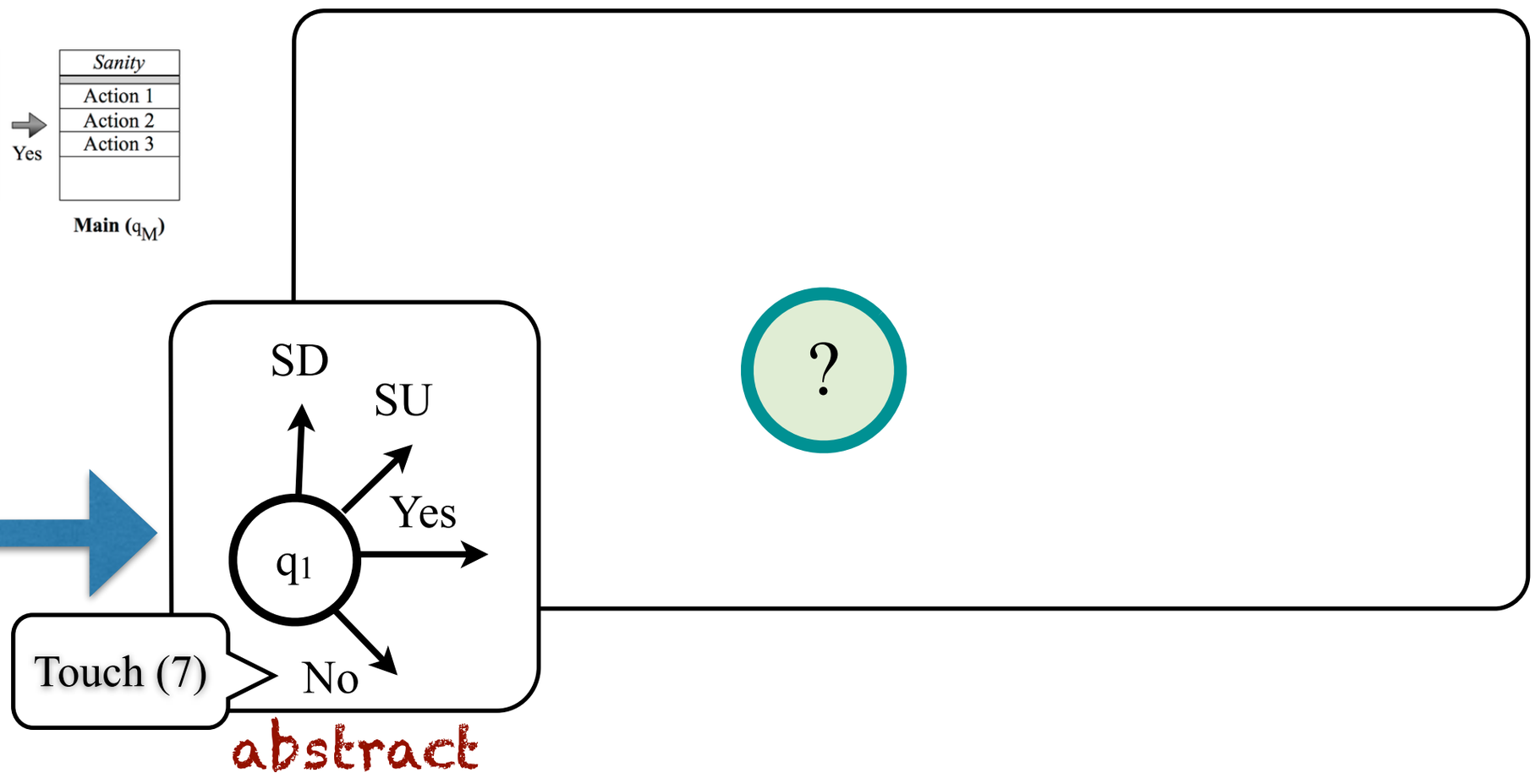
Actual Application





get GUI tree

Model Learned



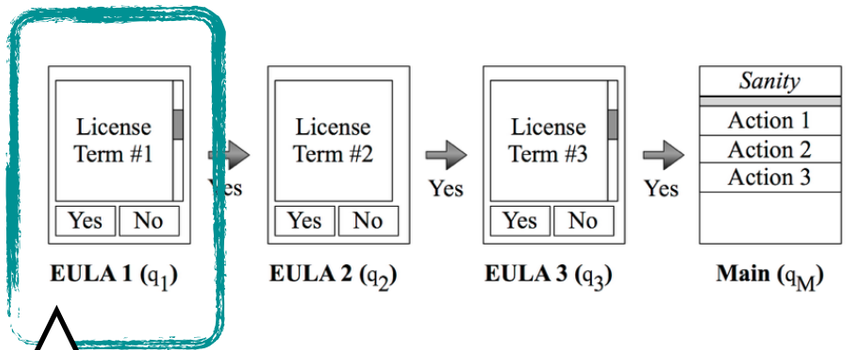
abstract

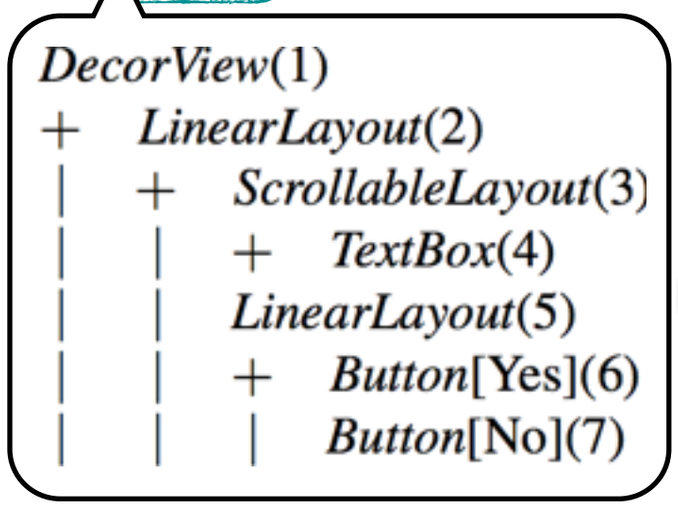
SwiftHand constructs a model
via observing the application screen.

SwiftHand with Example

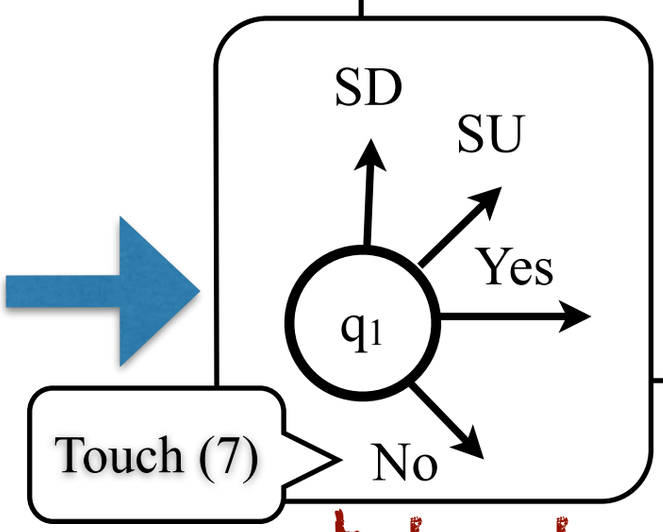
Initialization

Actual Application



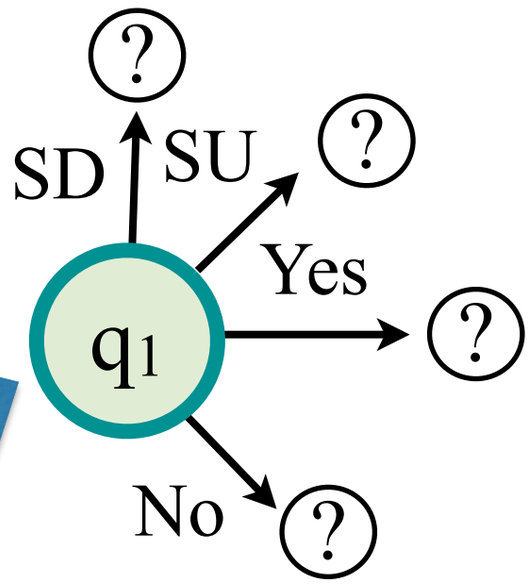


get GUI tree



abstract

Model Learned



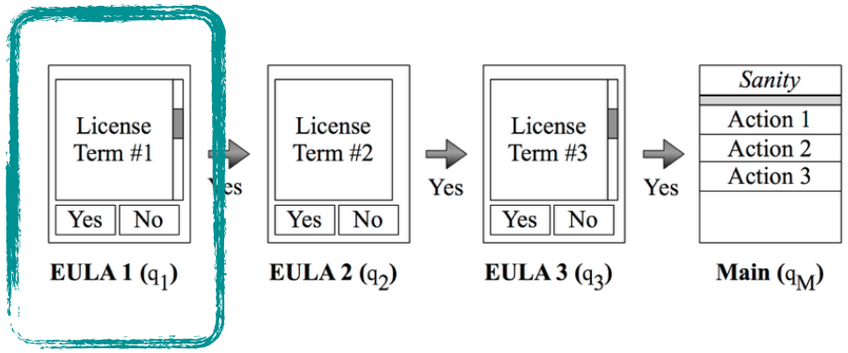
refine

SwiftHand constructs a model
via observing the application screen.

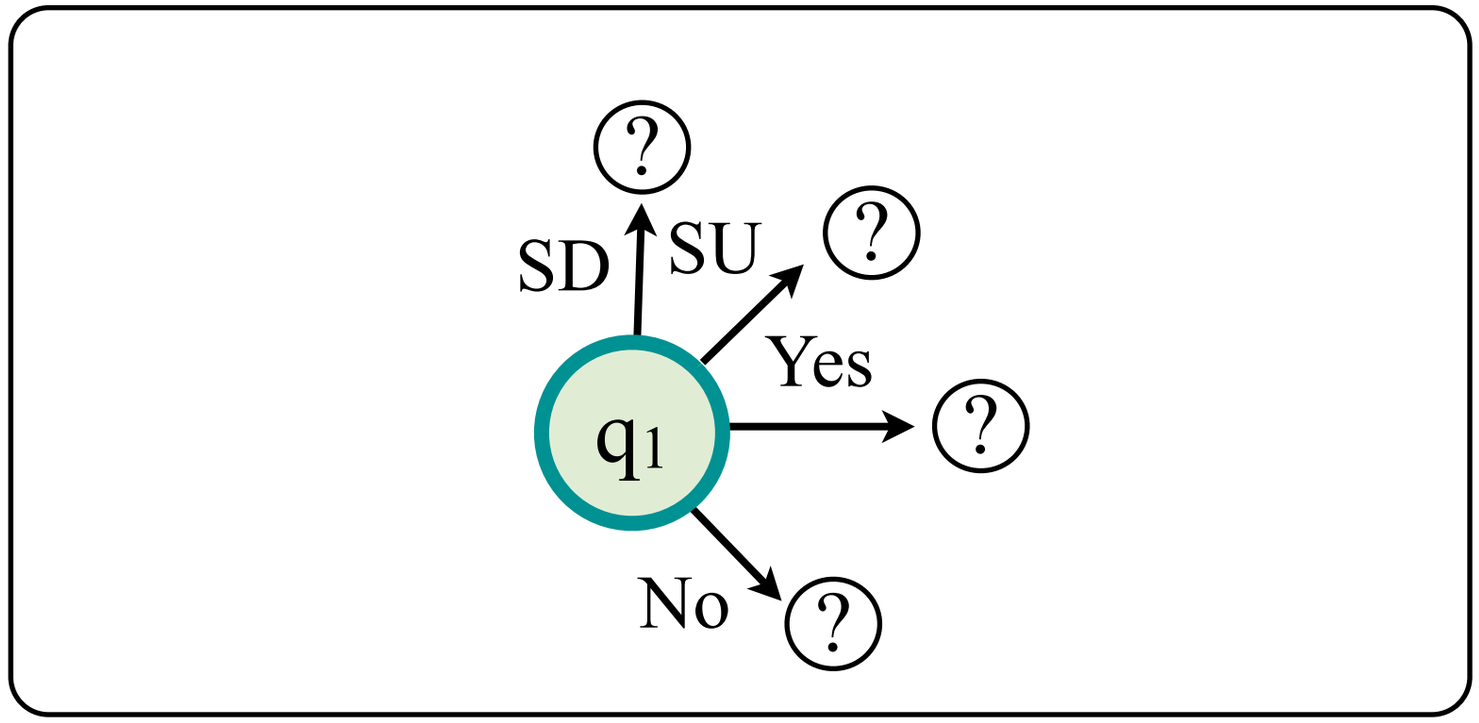
SwiftHand with Example

Iteration 1

Actual Application



Model Learned

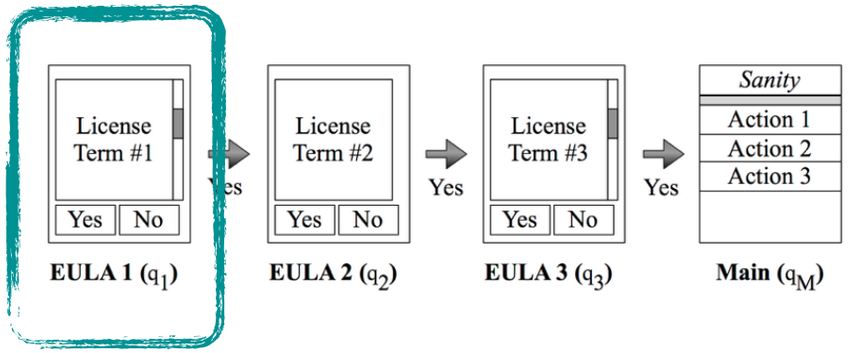


pick a target

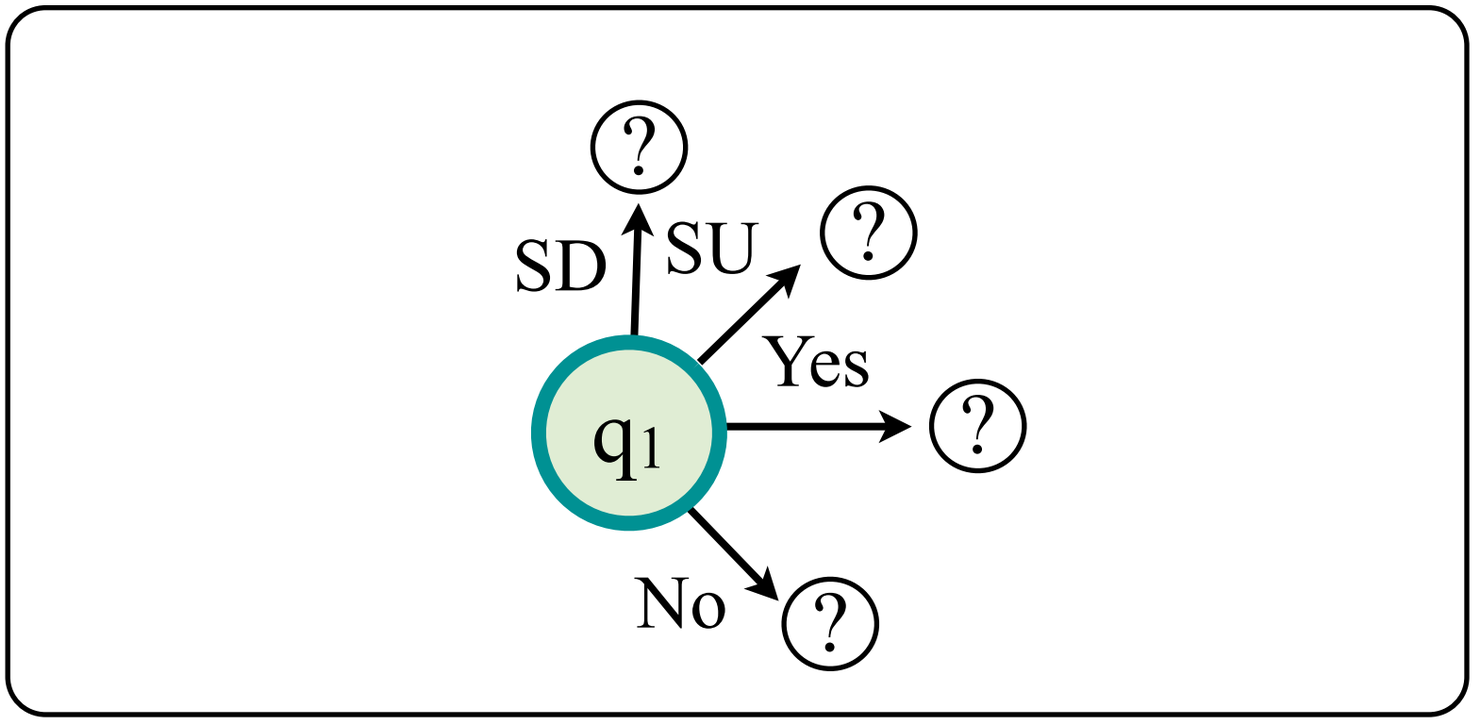
SwiftHand with Example

Iteration 1

Actual Application



Model Learned



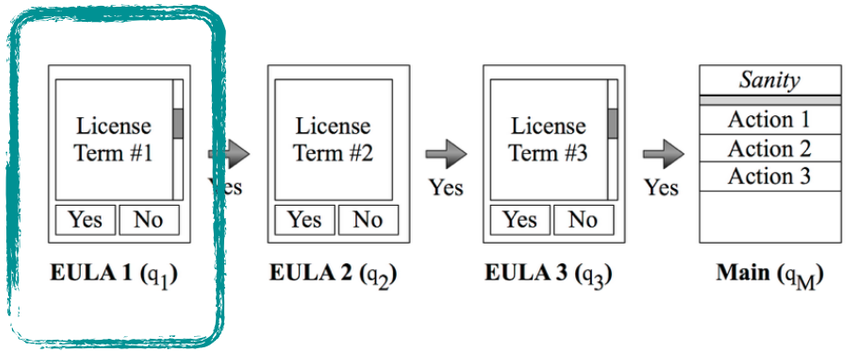
pick a target

SwiftHand decides the next state to visit based on the learned partial model.

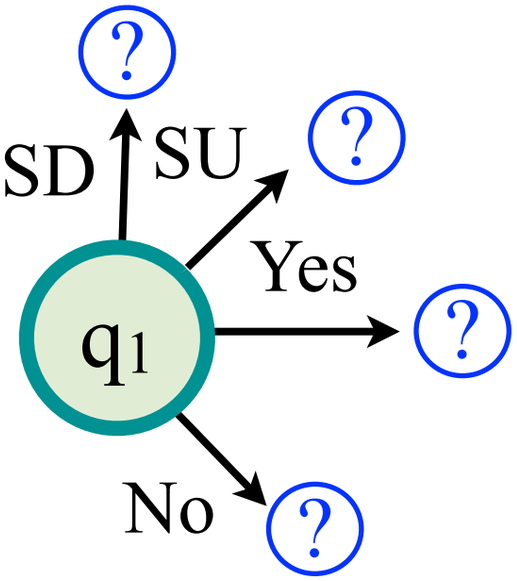
SwiftHand with Example

Iteration 1

Actual Application



Model Learned



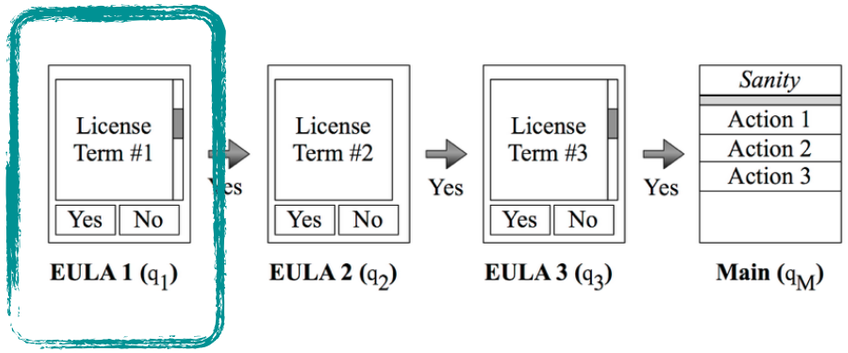
pick a target

SwiftHand decides the next state to visit based on the learned partial model.

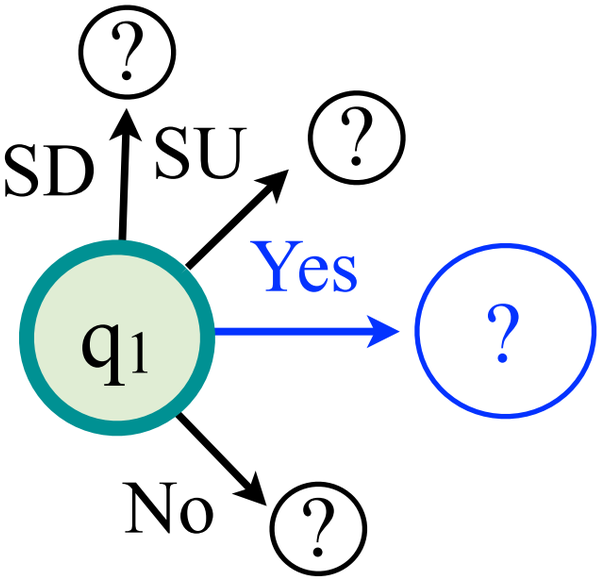
SwiftHand with Example

Iteration 1

Actual Application



Model Learned



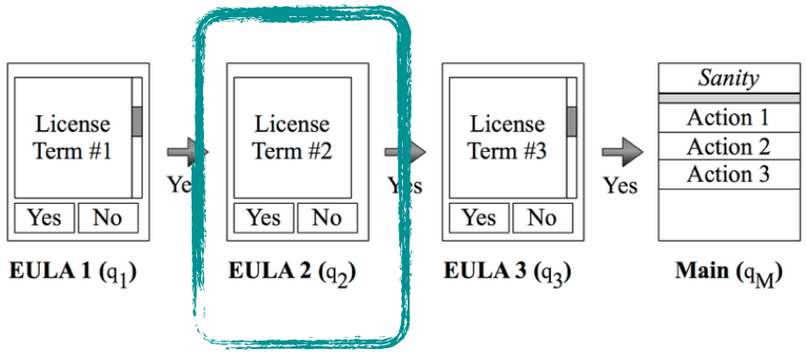
pick a target

SwiftHand decides the next state to visit based on the learned partial model.

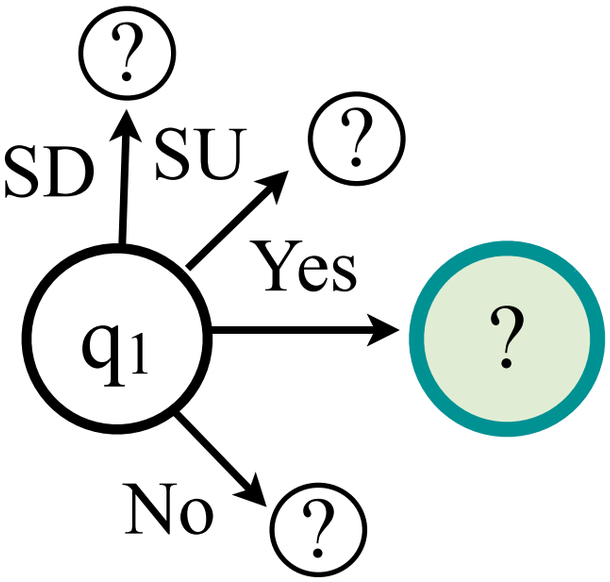
SwiftHand with Example

Iteration 1

Actual Application



Model Learned



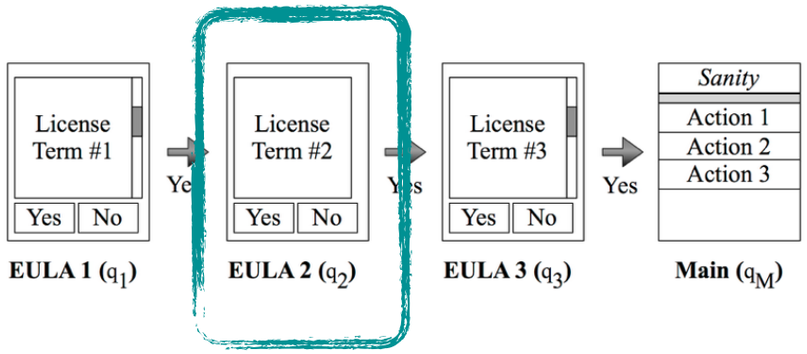
execute

SwiftHand decides the next state to visit based on the learned partial model.

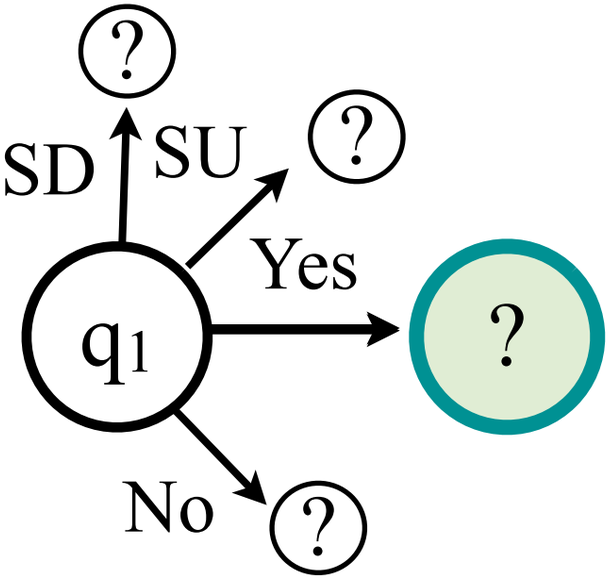
SwiftHand with Example

Iteration 1

Actual Application



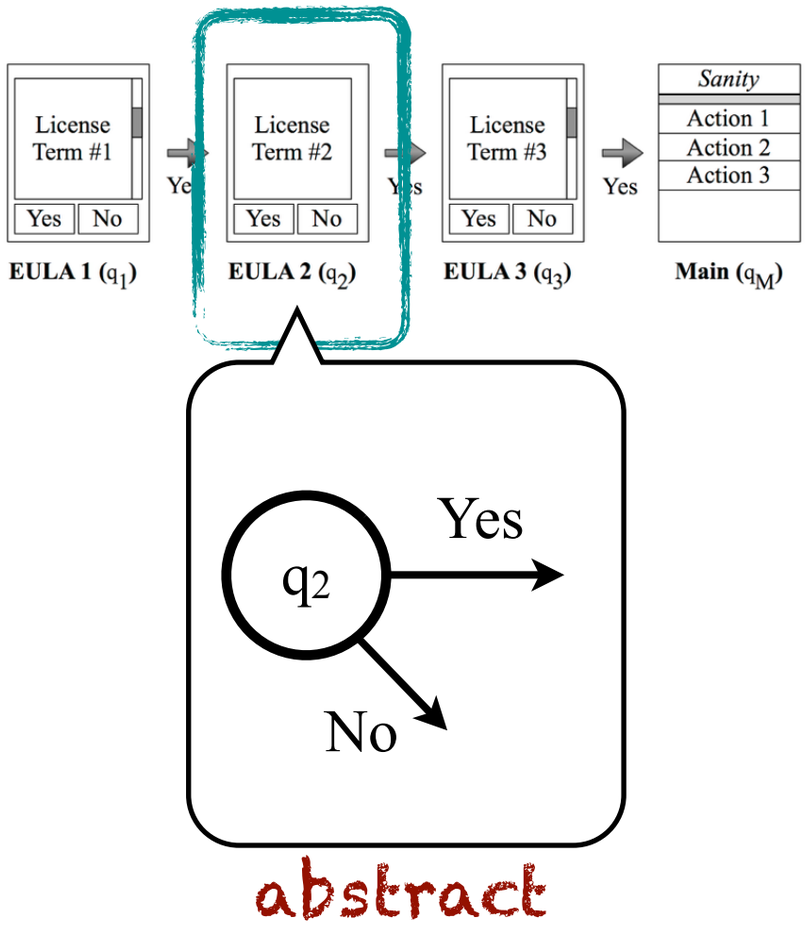
Model Learned



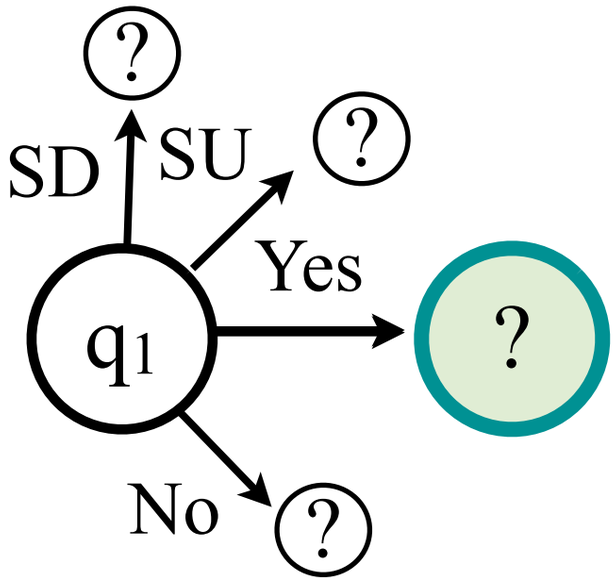
SwiftHand with Example

Iteration 1

Actual Application



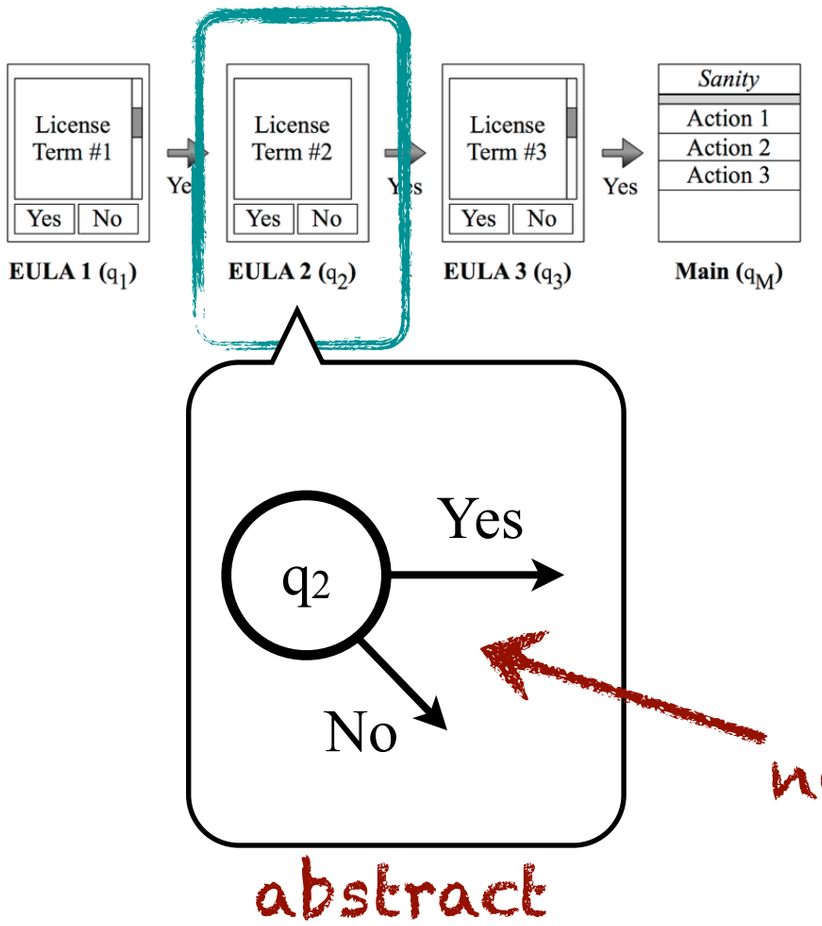
Model Learned



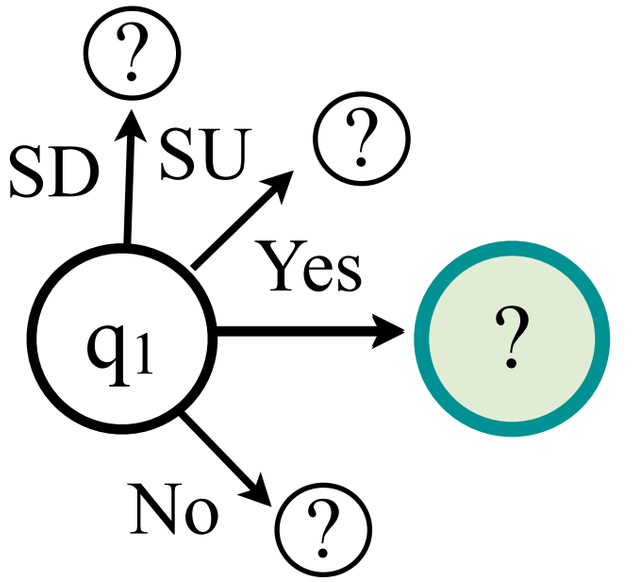
SwiftHand with Example

Iteration 1

Actual Application



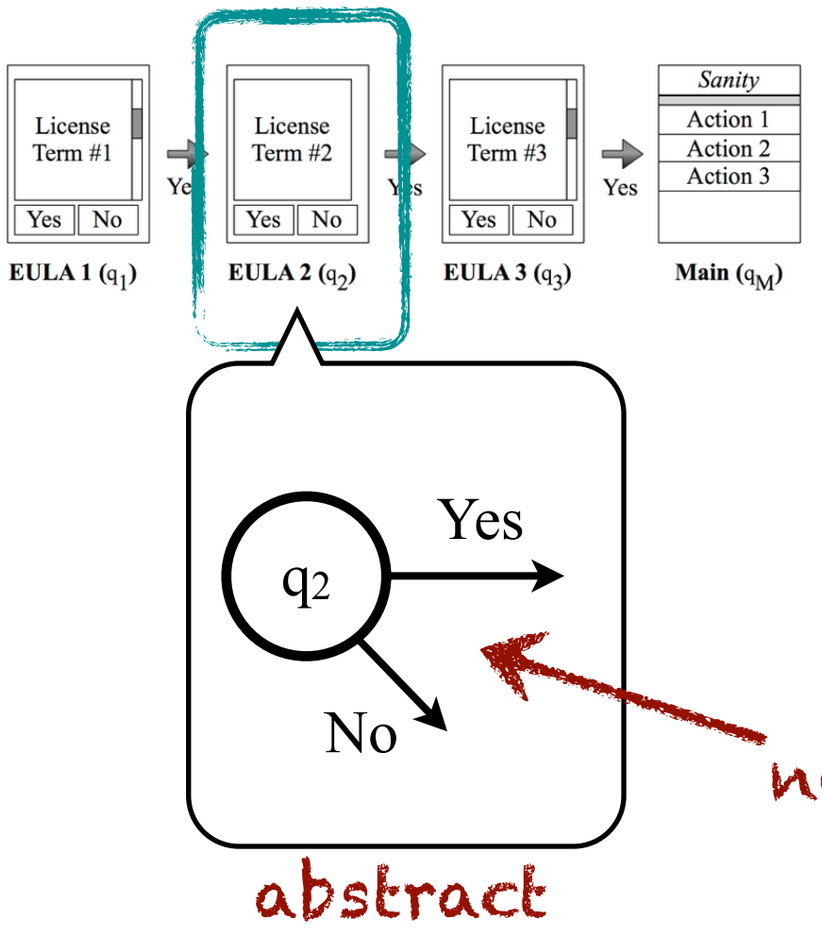
Model Learned



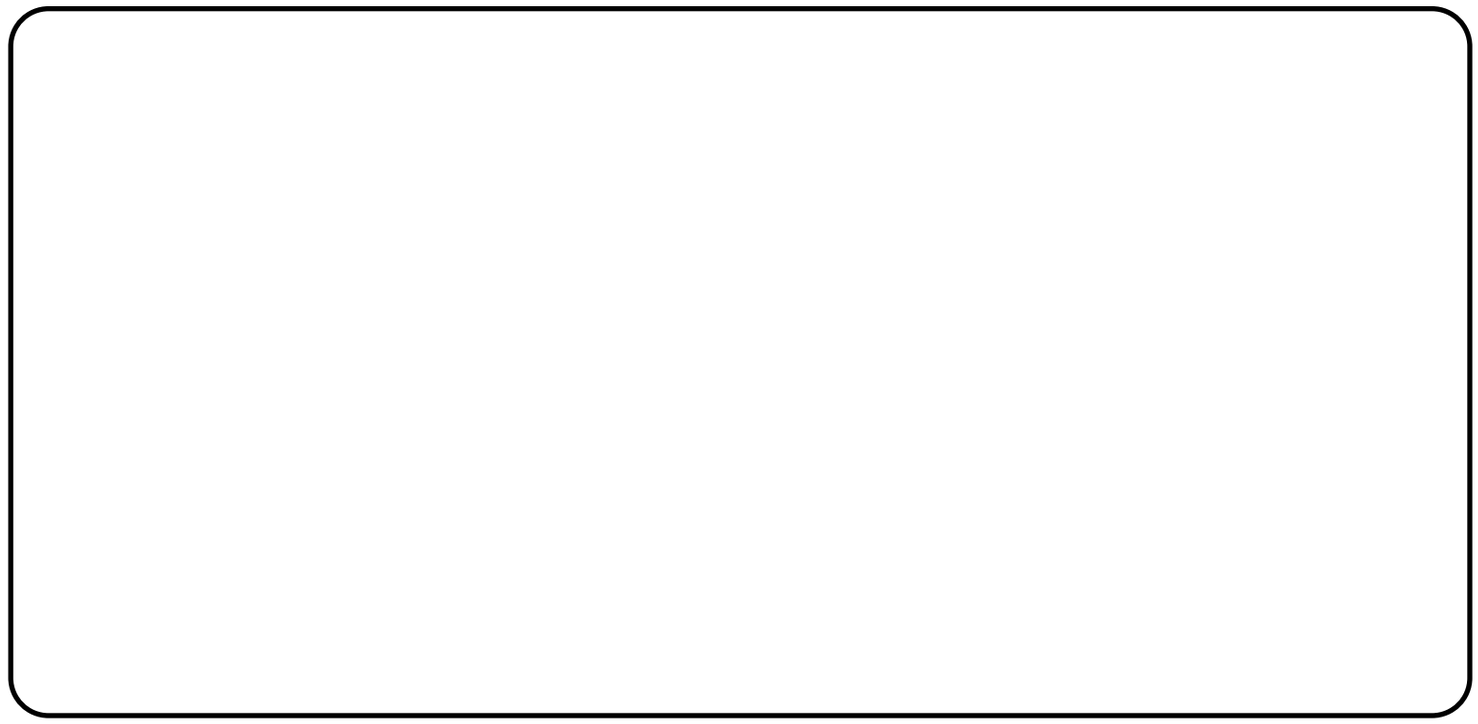
SwiftHand with Example

Iteration 1

Actual Application



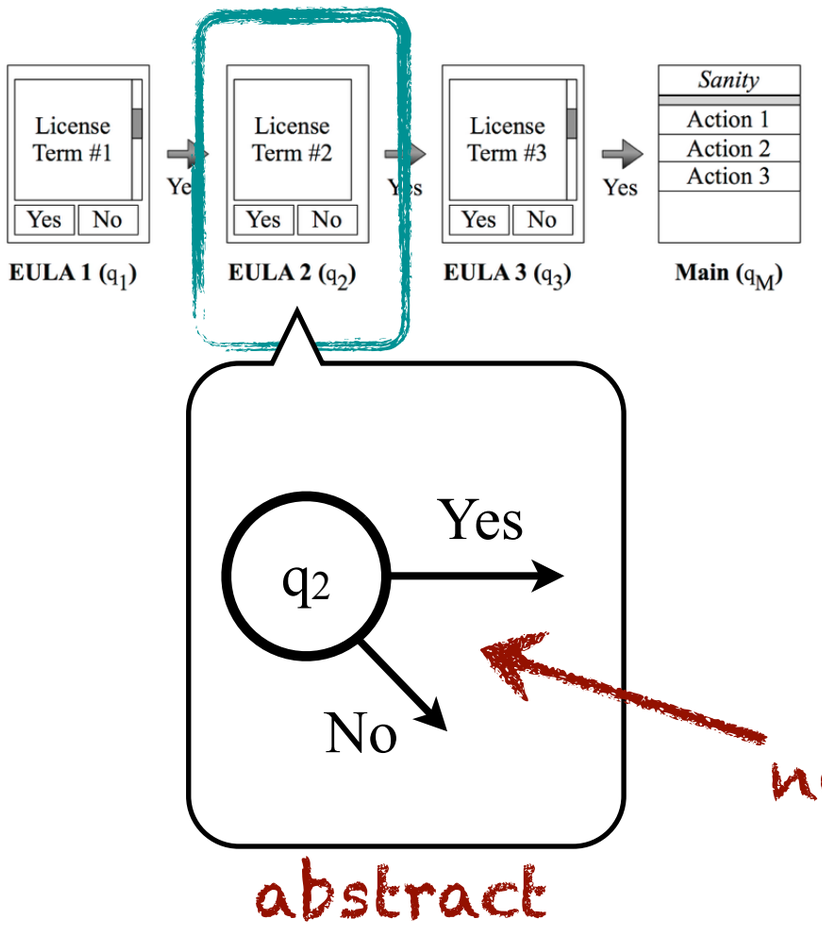
Model Learned



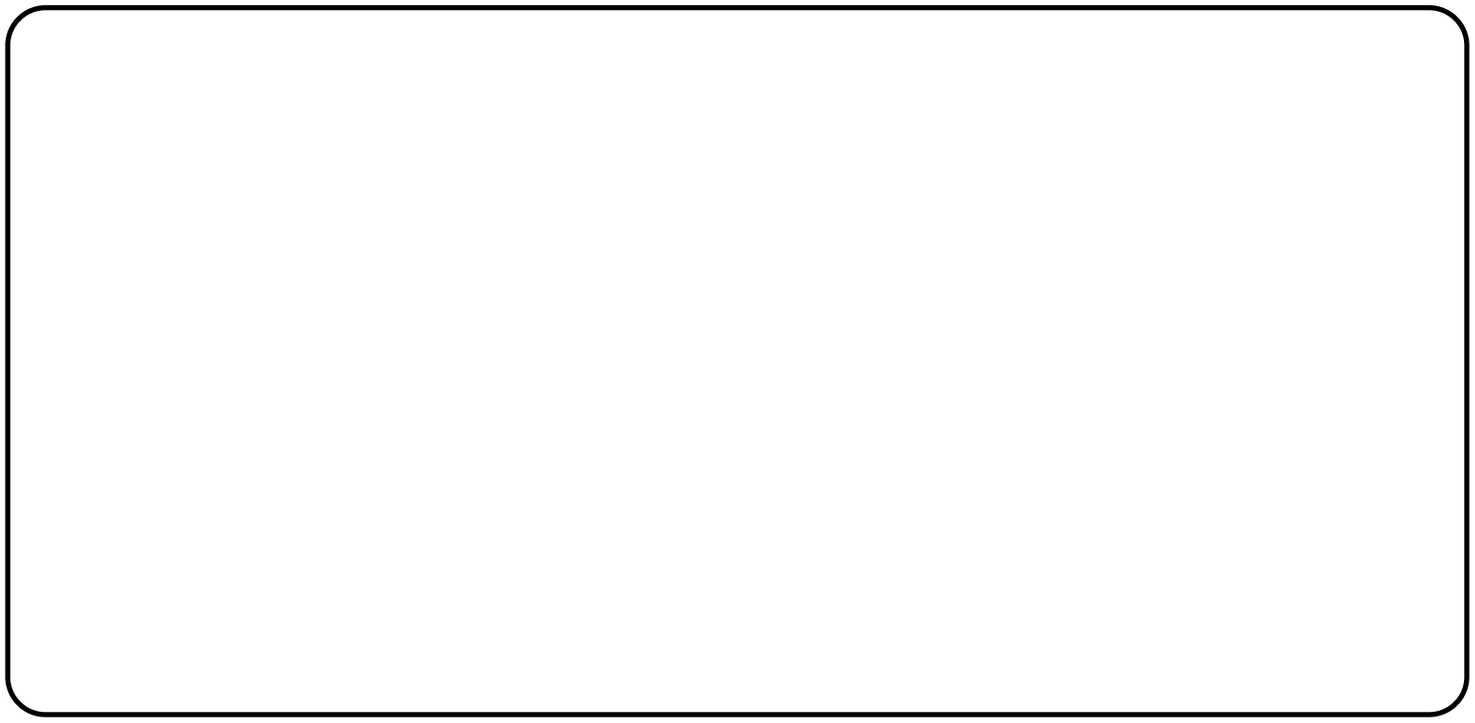
SwiftHand with Example

Iteration 1

Actual Application



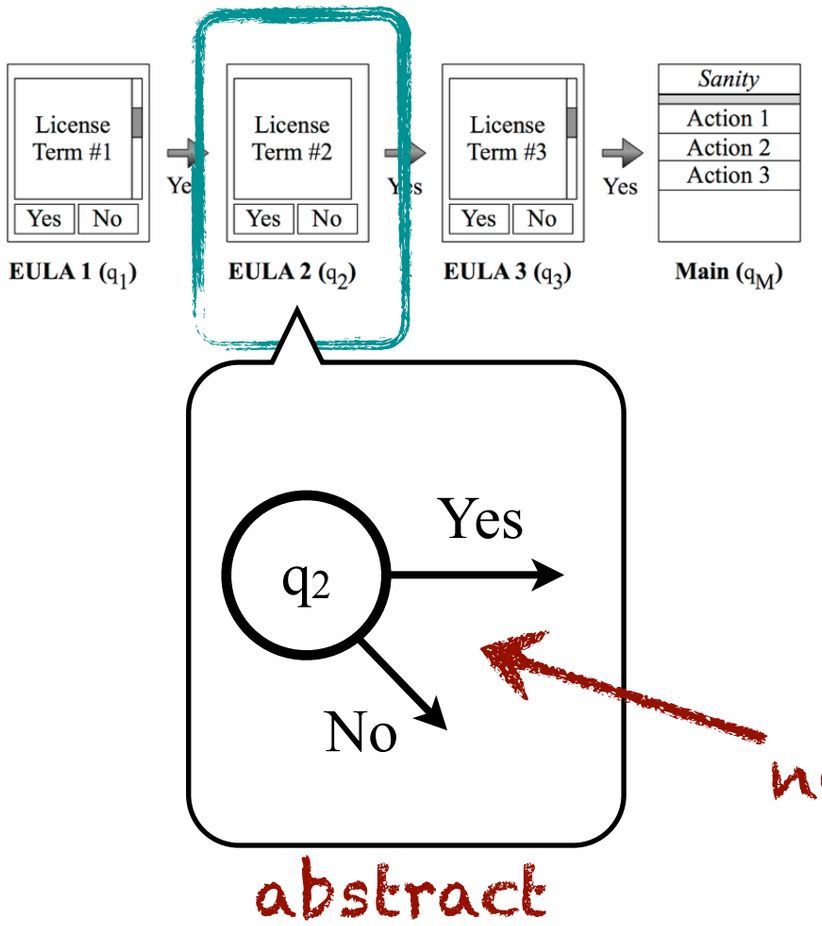
Model Learned



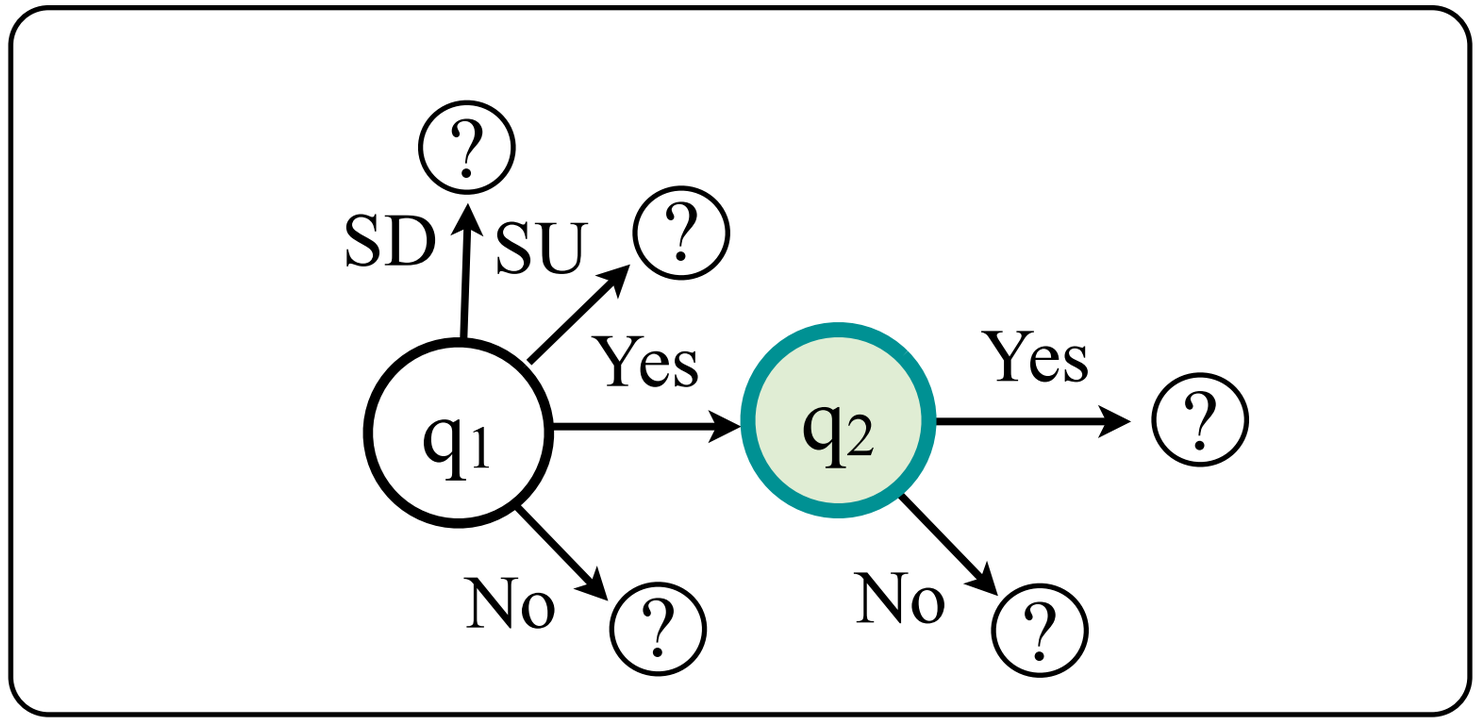
SwiftHand with Example

Iteration 1

Actual Application



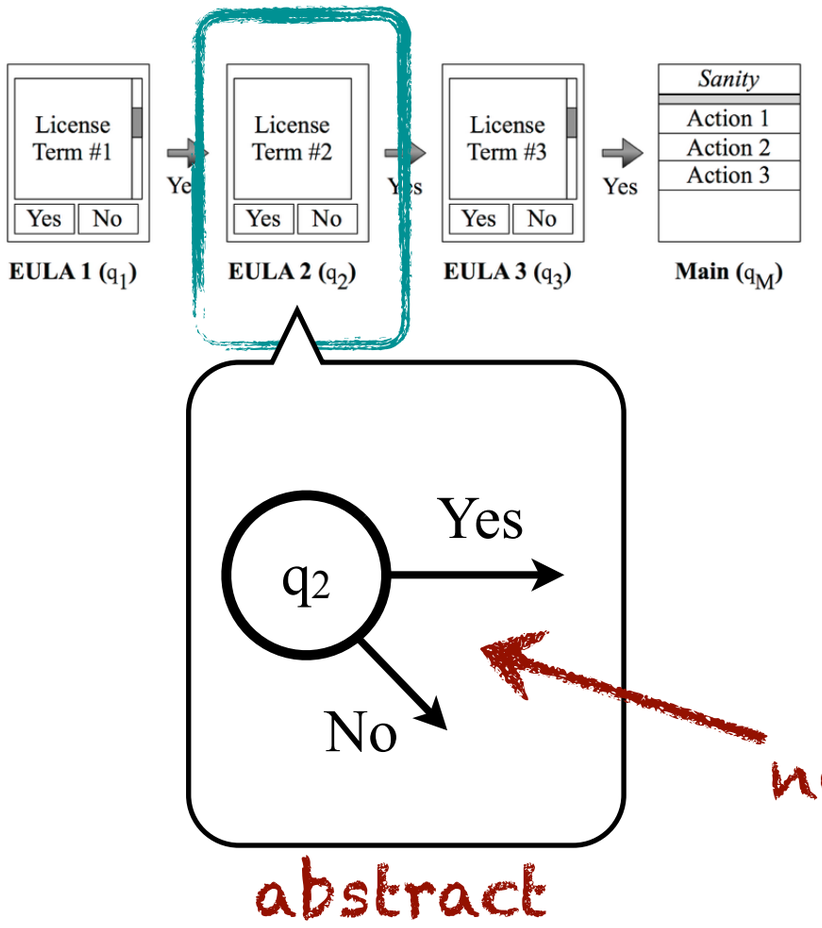
Model Learned



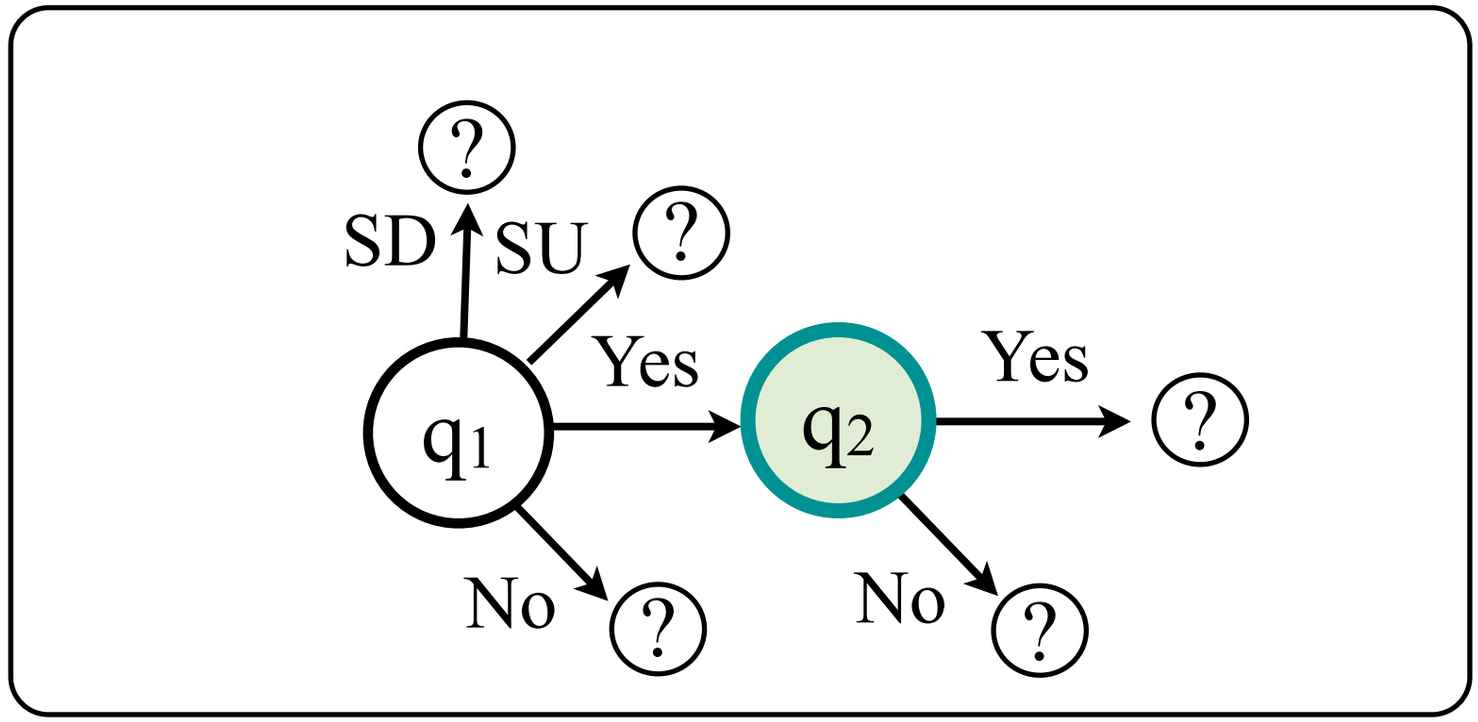
SwiftHand with Example

Iteration 1

Actual Application



Model Learned

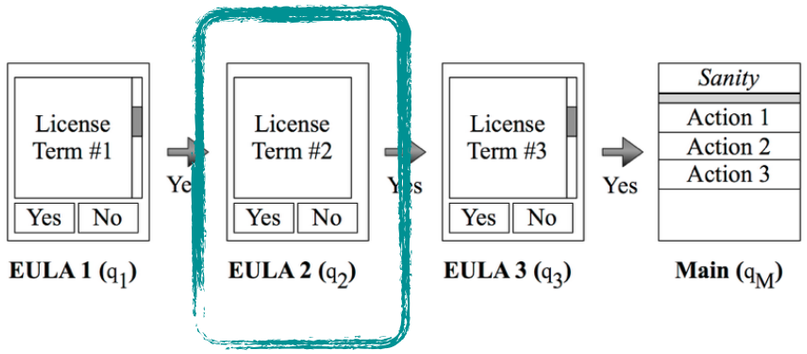


The new screen has different enabled inputs.
Add a new state to the model

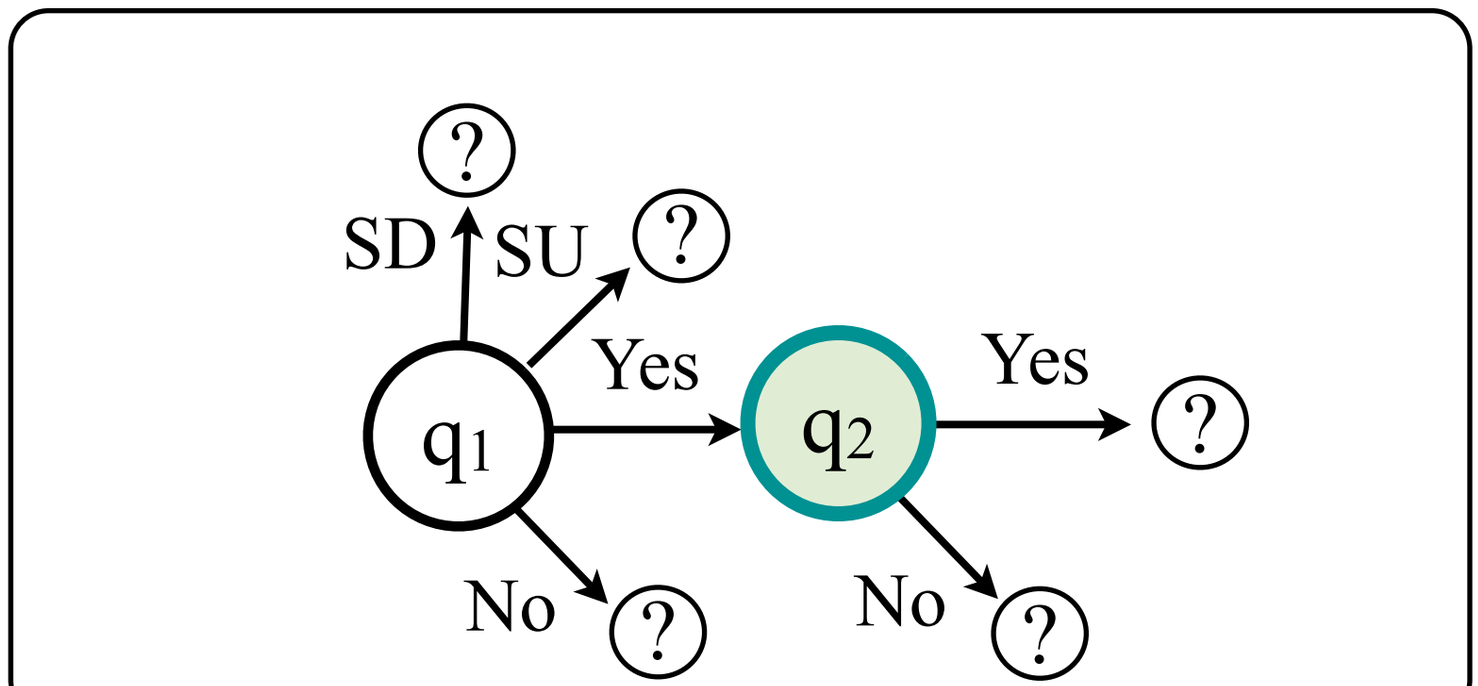
SwiftHand with Example

Iteration 2

Actual Application



Model Learned

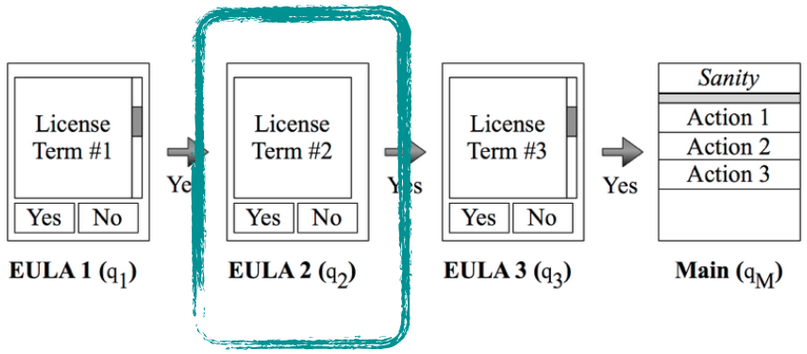


pick a target

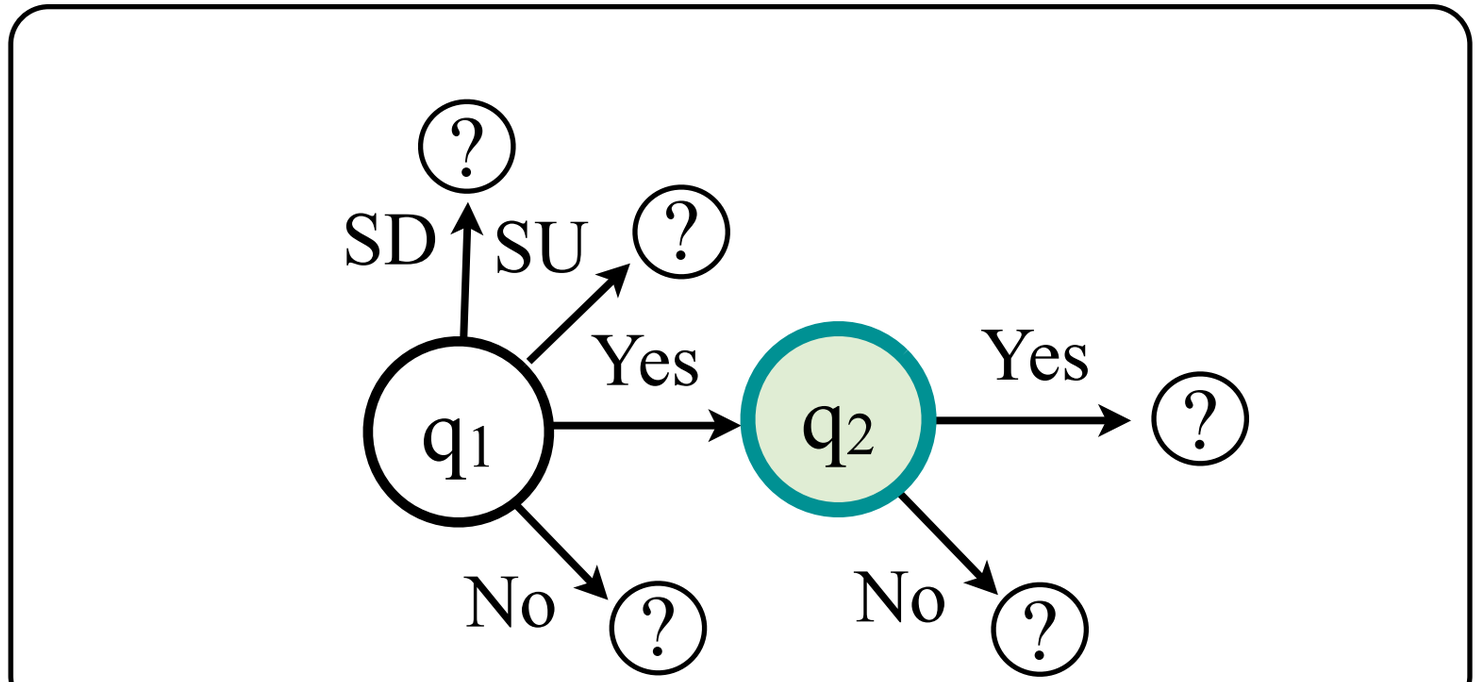
SwiftHand with Example

Iteration 2

Actual Application



Model Learned



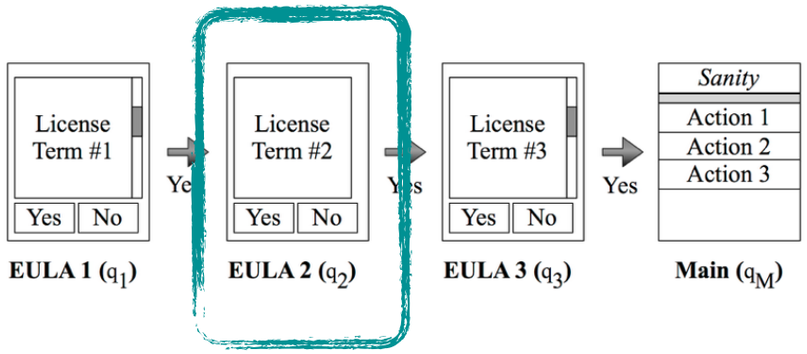
pick a target

Chooses a target state
reachable from the current model state (avoid restart)

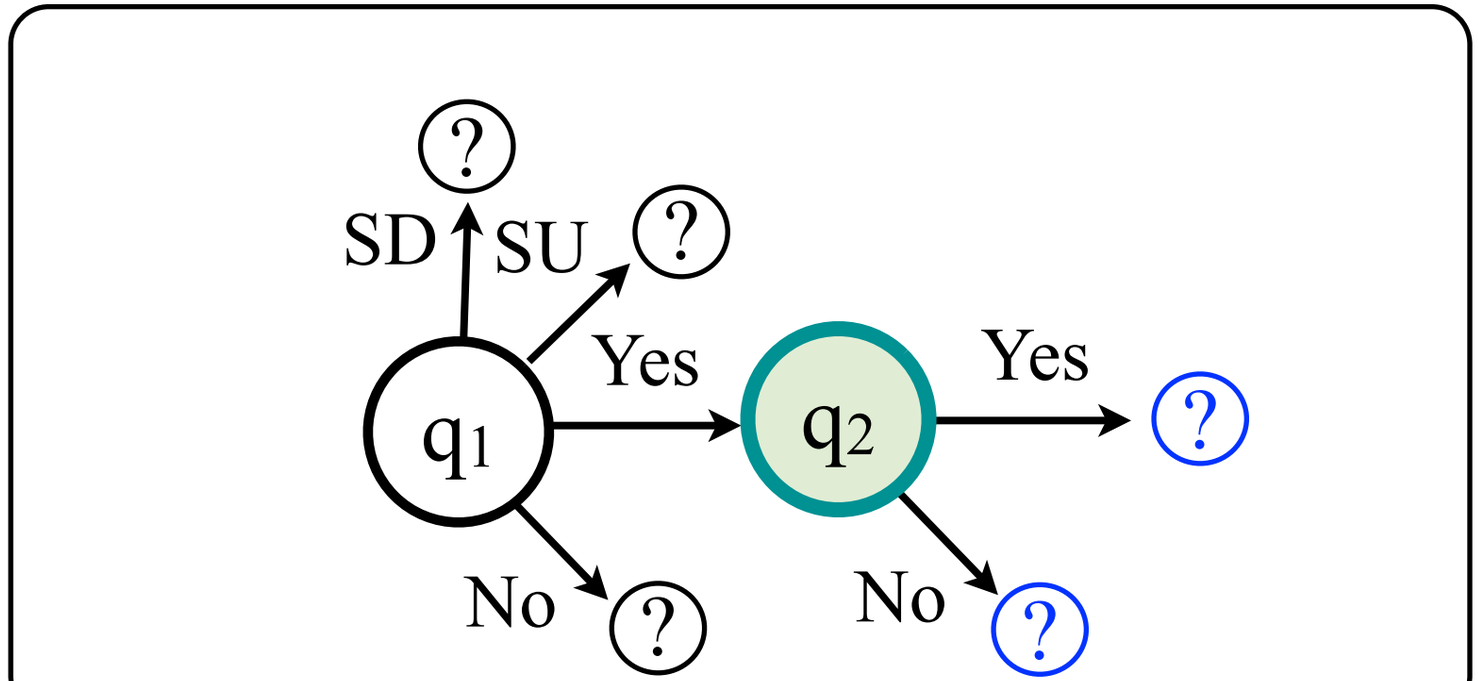
SwiftHand with Example

Iteration 2

Actual Application



Model Learned



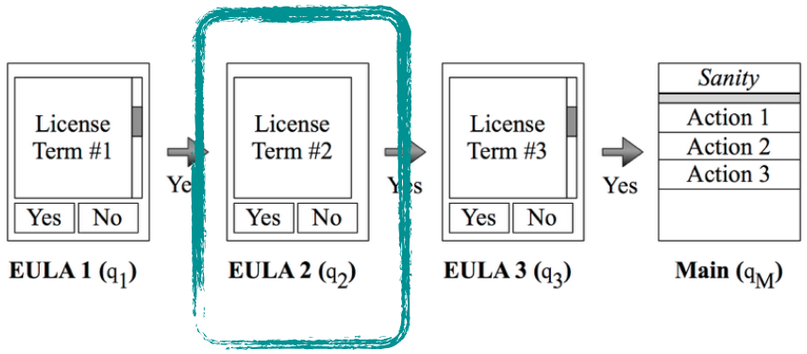
pick a target

Chooses a target state
reachable from the current model state (avoid restart)

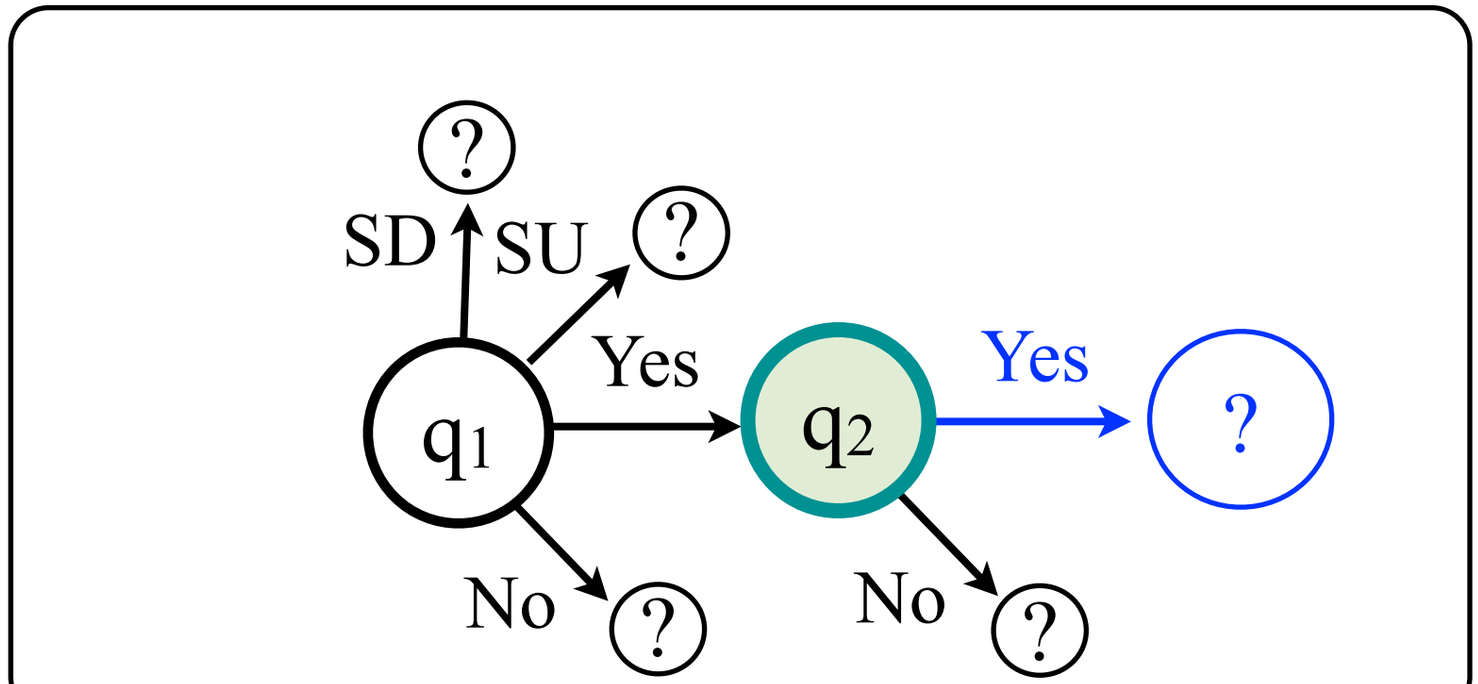
SwiftHand with Example

Iteration 2

Actual Application



Model Learned



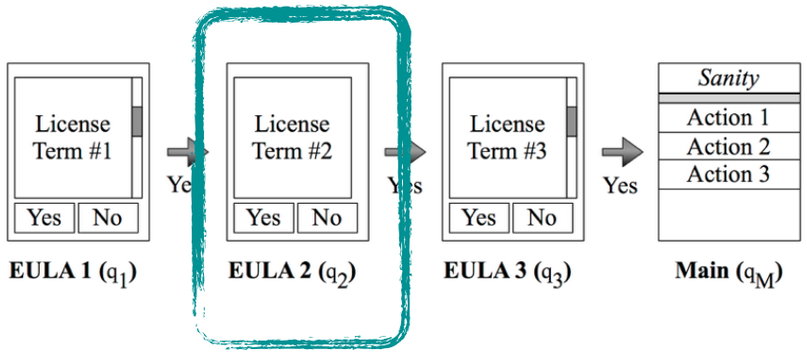
pick a target

Chooses a target state
reachable from the current model state (avoid restart)

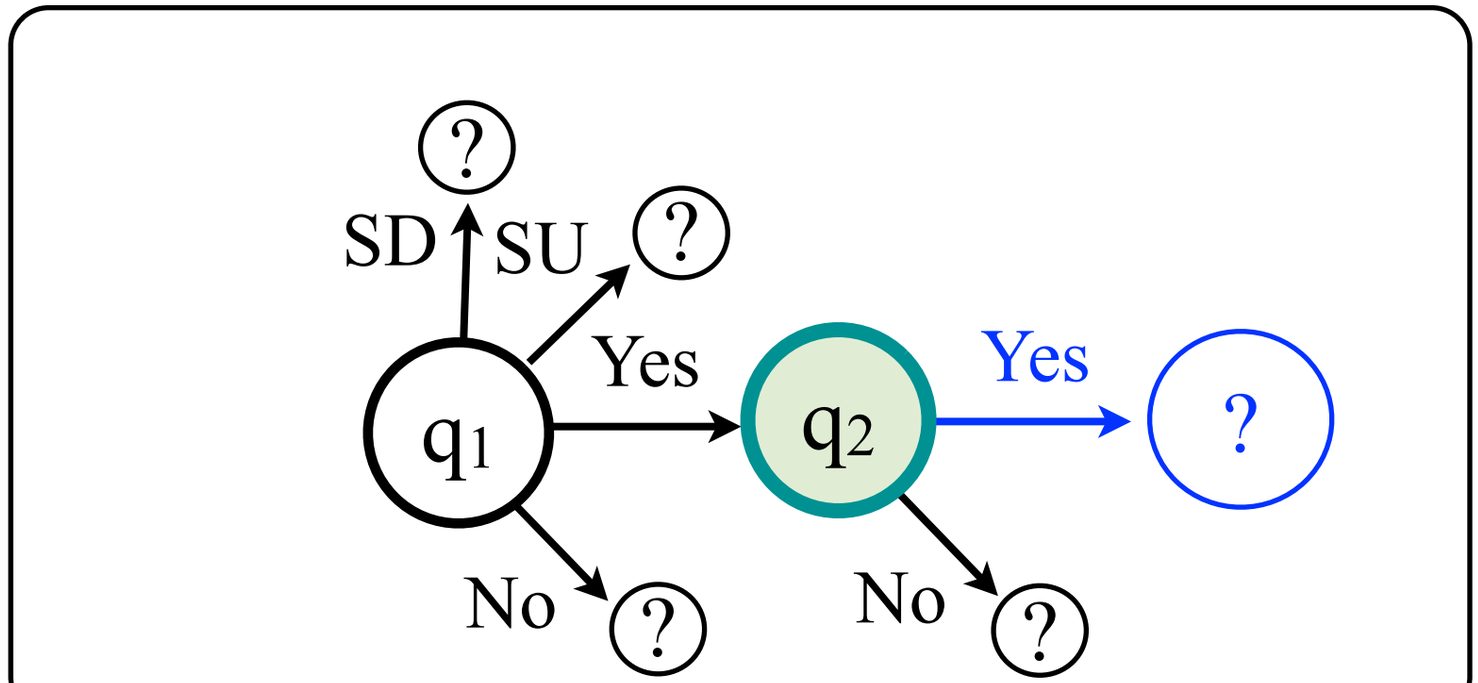
SwiftHand with Example

Iteration 2

Actual Application



Model Learned

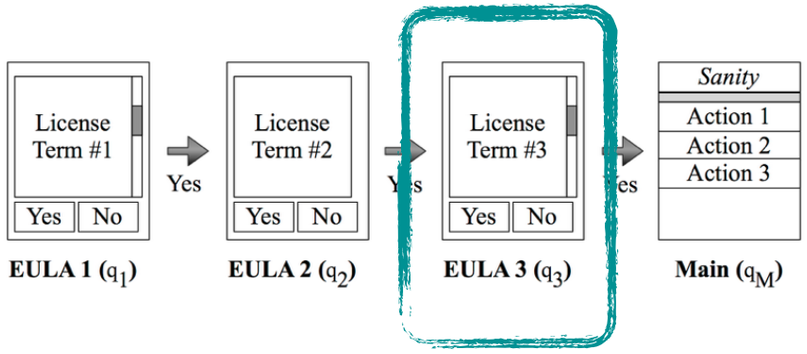


pick a target

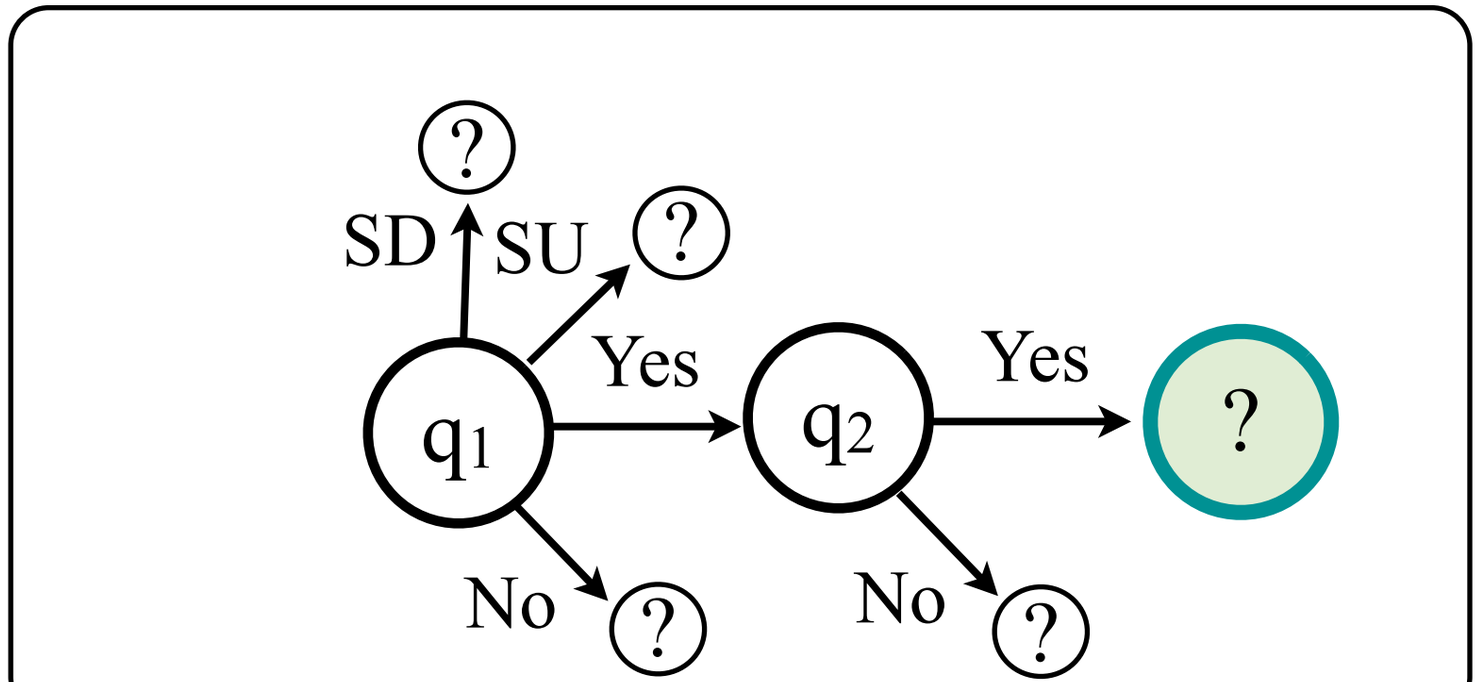
SwiftHand with Example

Iteration 2

Actual Application



Model Learned

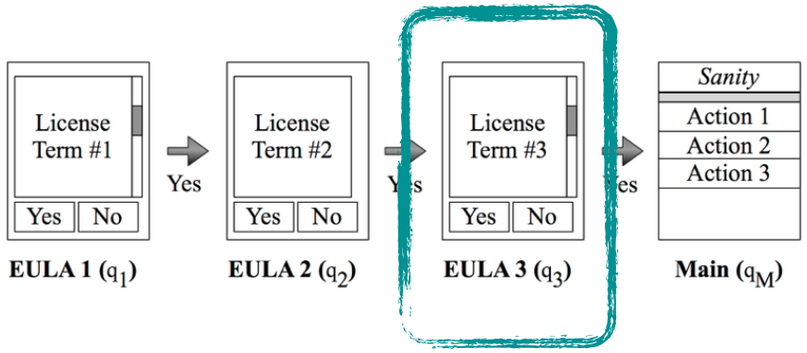


execute

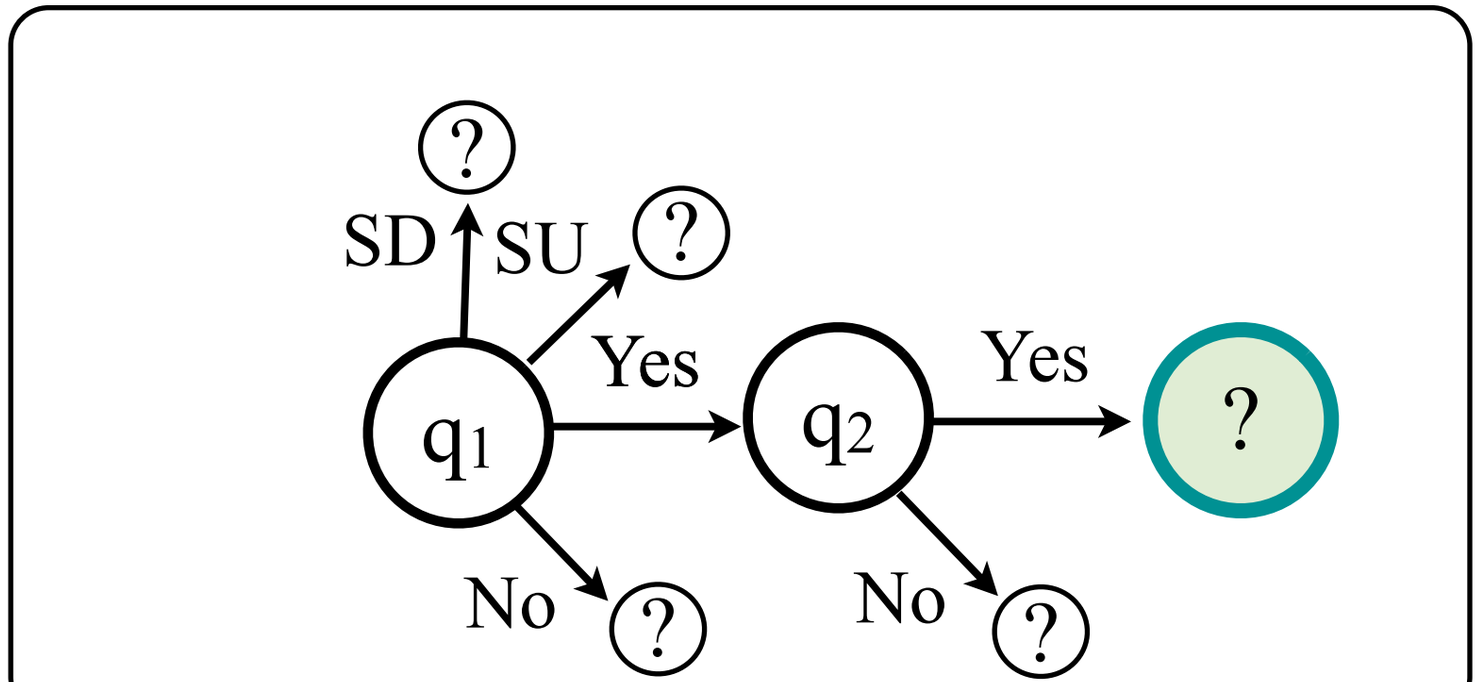
SwiftHand with Example

Iteration 2

Actual Application



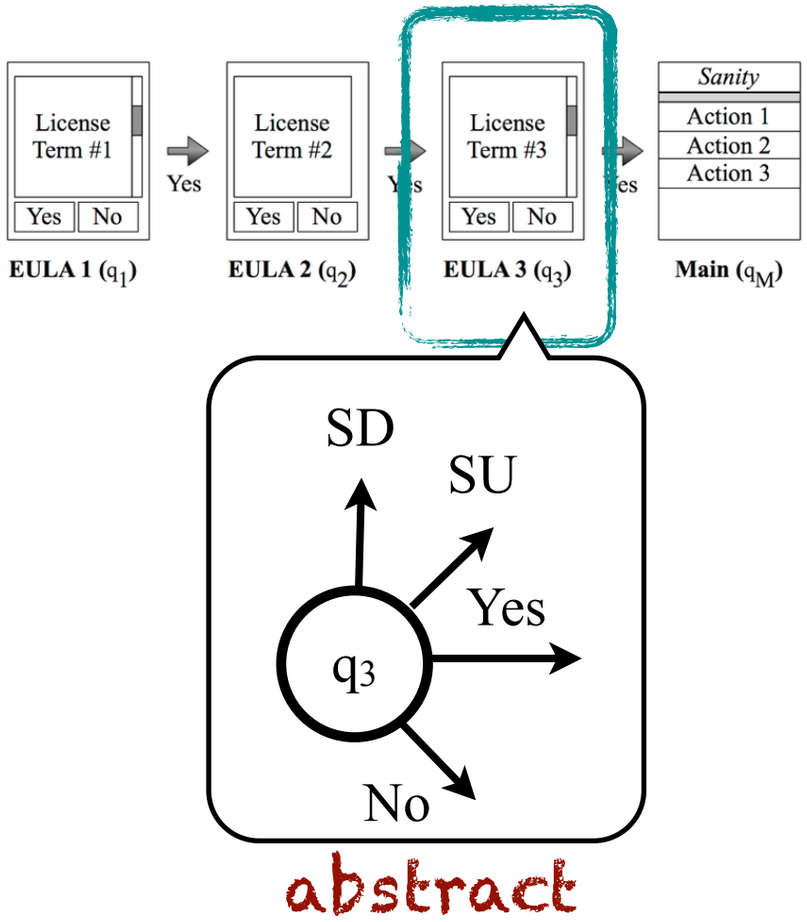
Model Learned



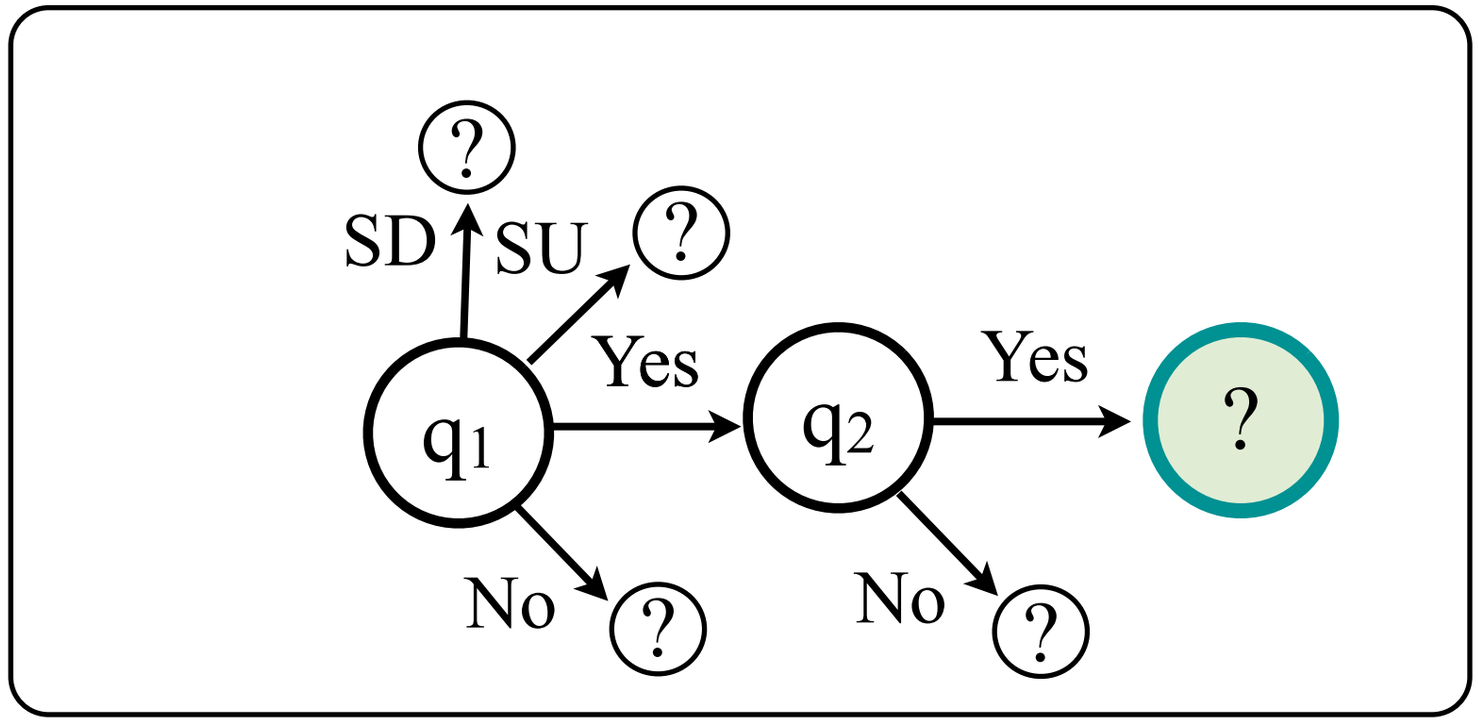
SwiftHand with Example

Iteration 2

Actual Application



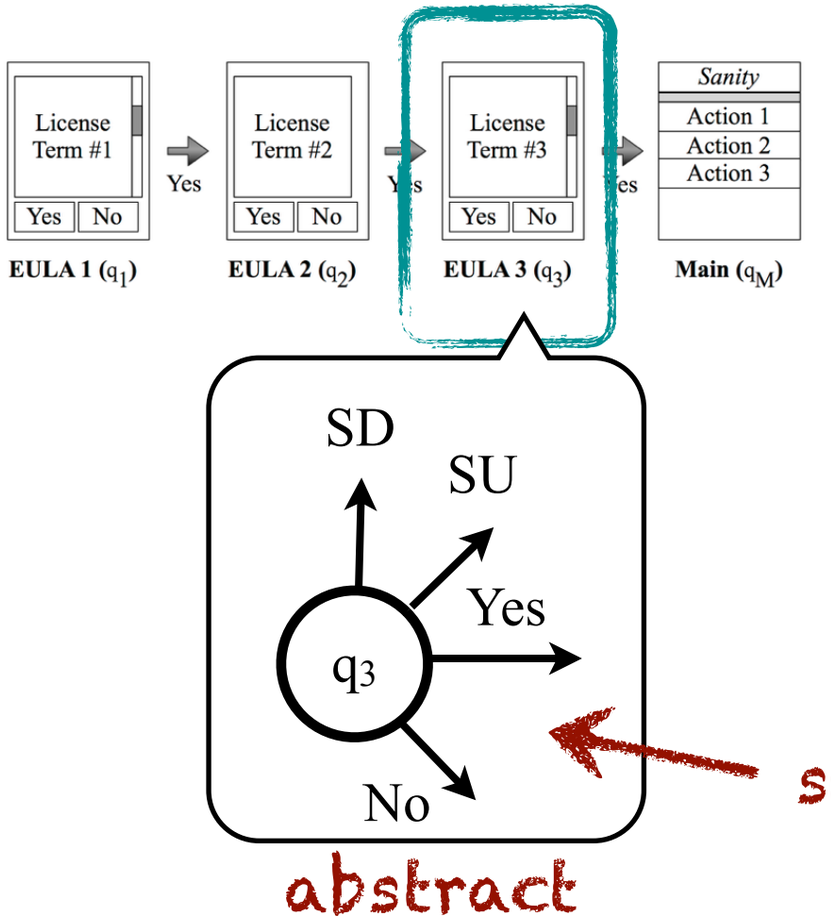
Model Learned



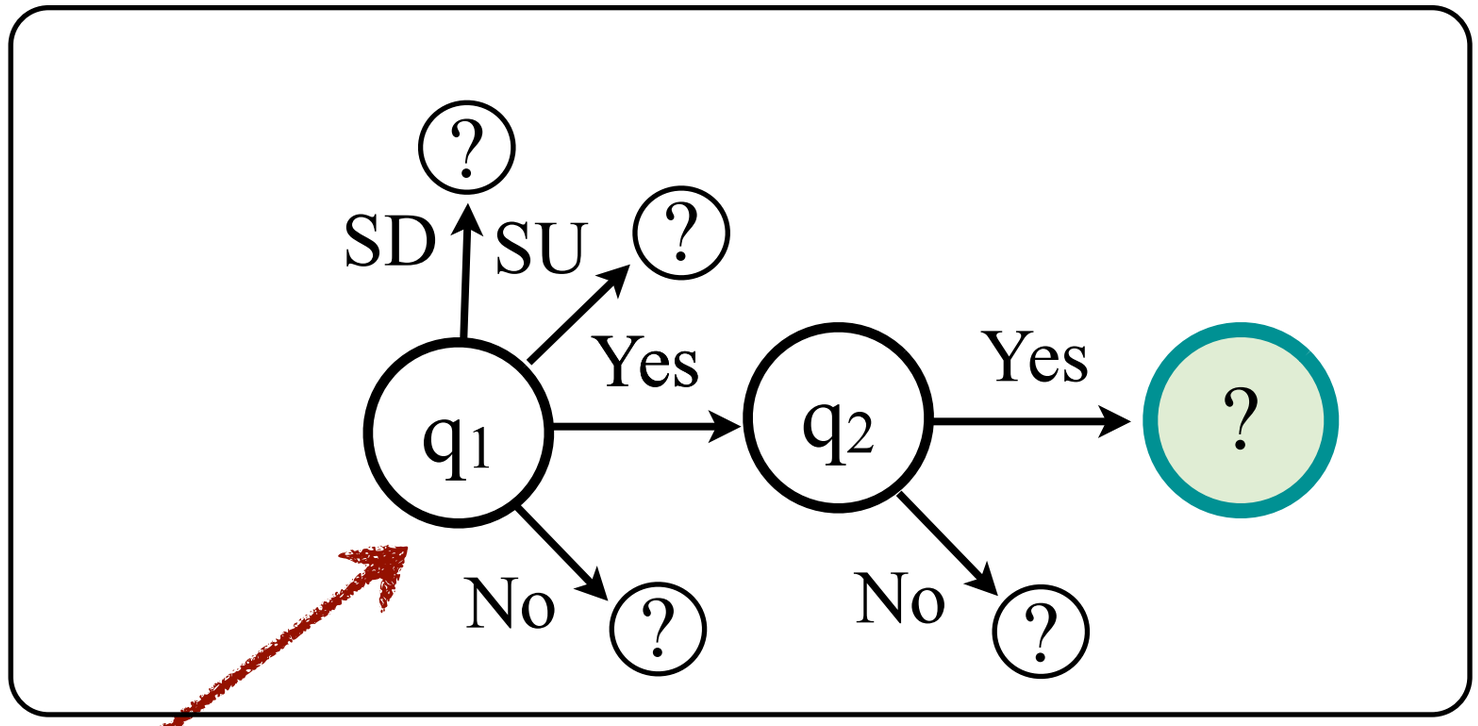
SwiftHand with Example

Iteration 2

Actual Application



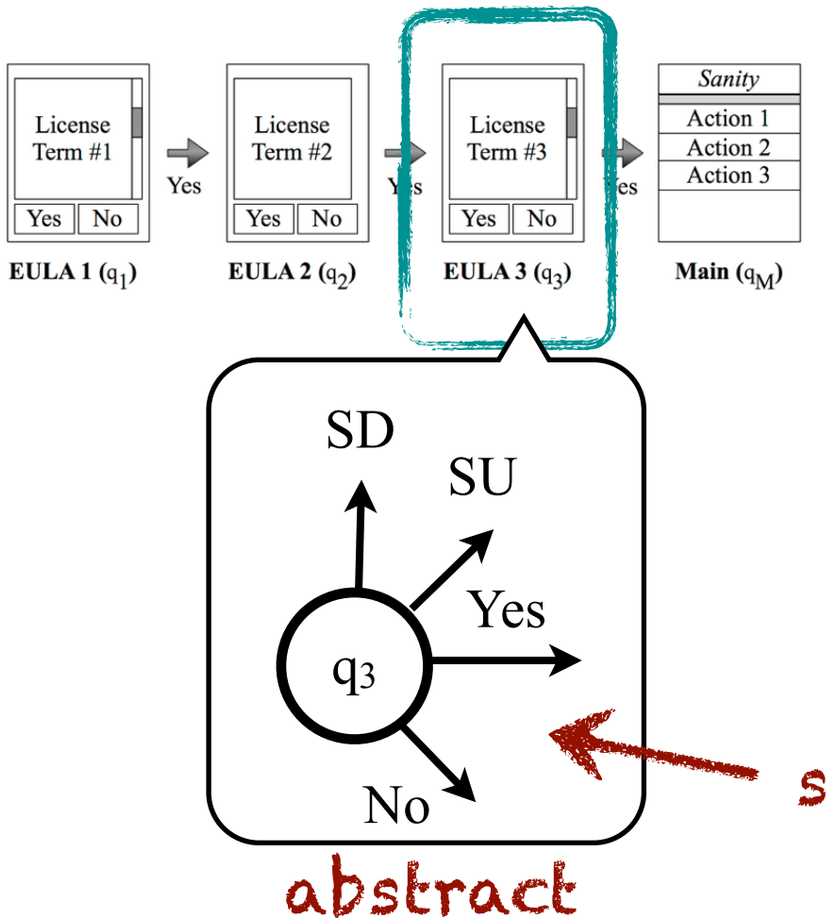
Model Learned



SwiftHand with Example

Iteration 2

Actual Application



Model Learned

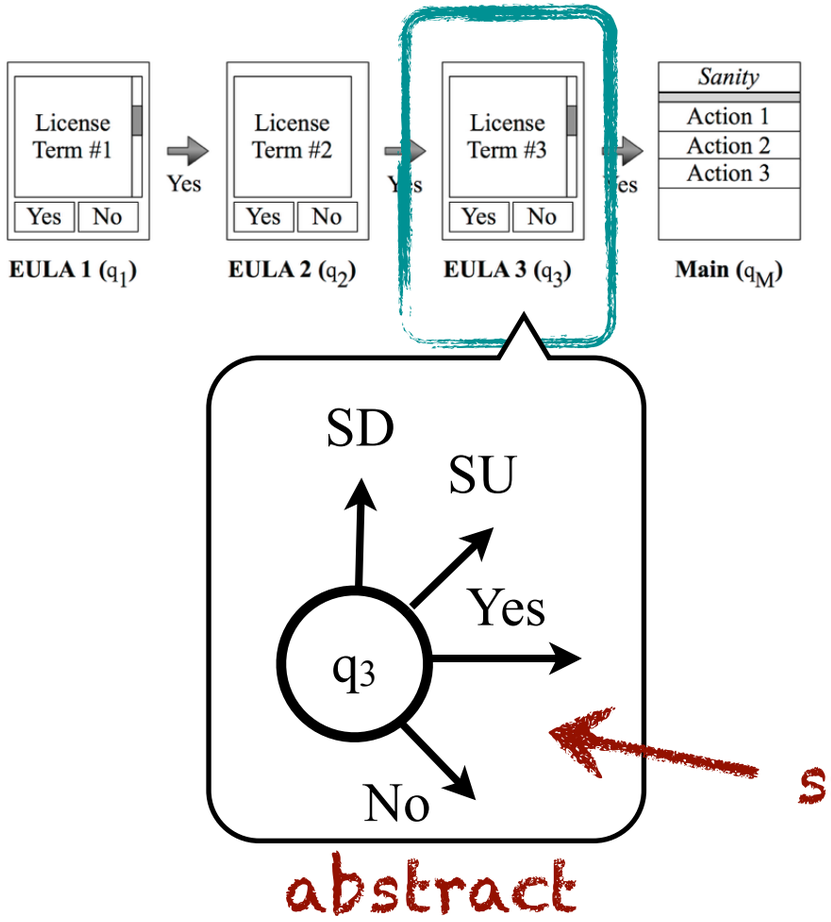
similar

merge similar states

SwiftHand with Example

Iteration 2

Actual Application



Model Learned

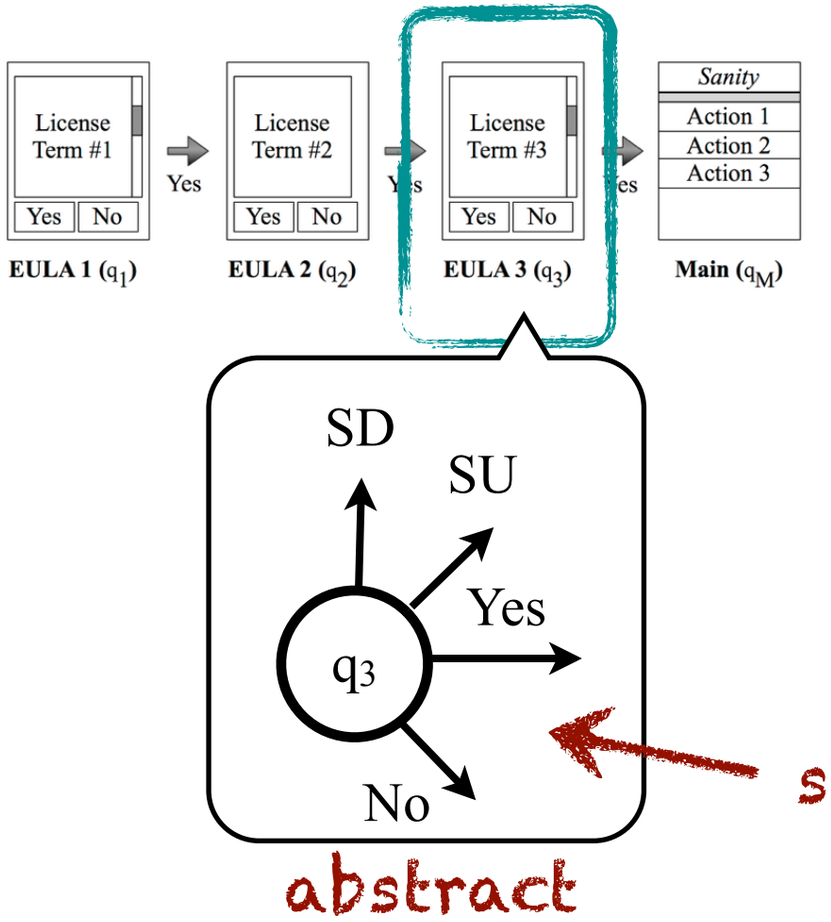
similar

merge similar states

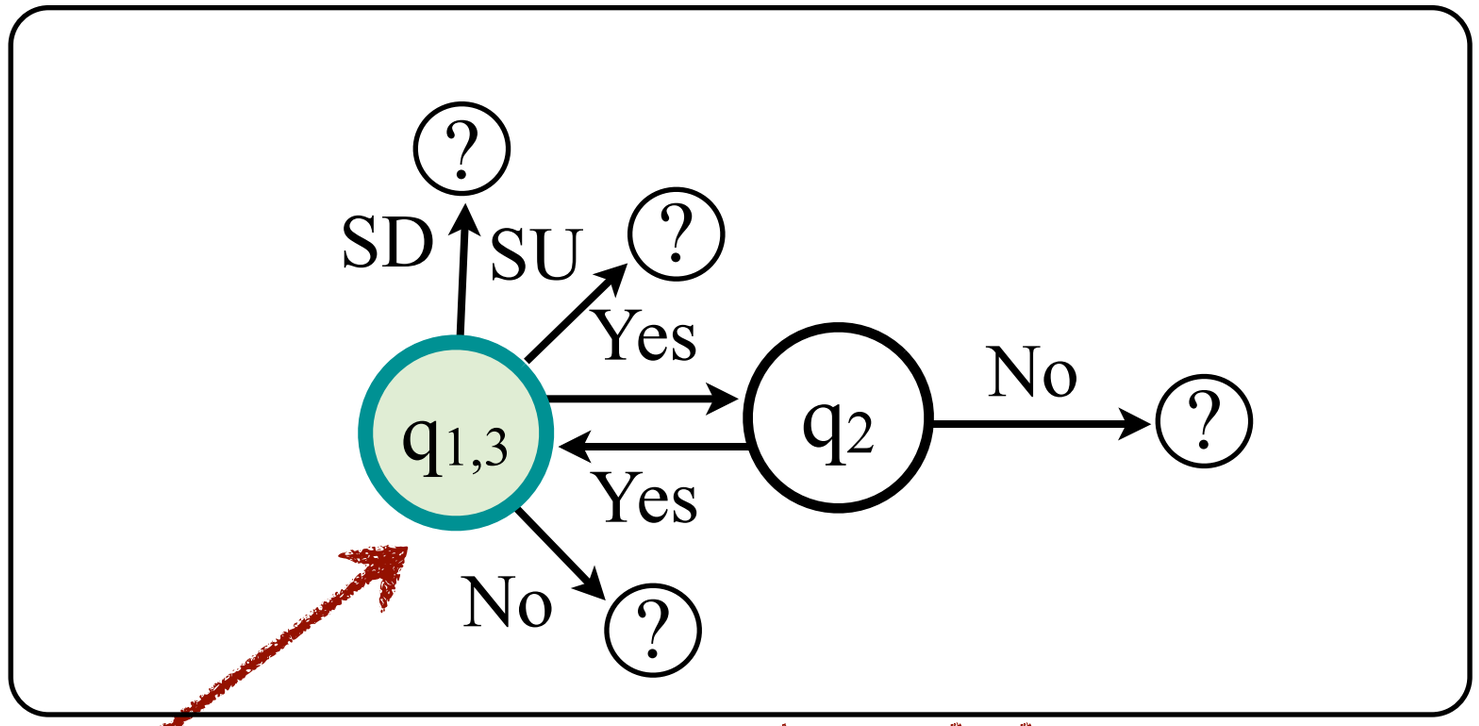
SwiftHand with Example

Iteration 2

Actual Application



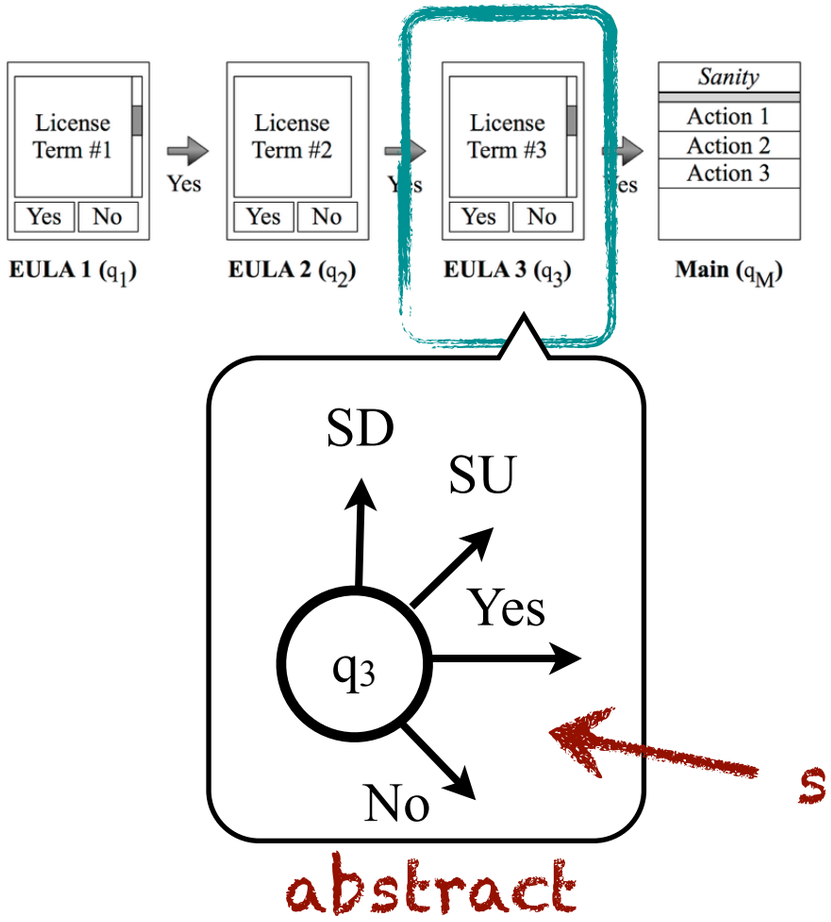
Model Learned



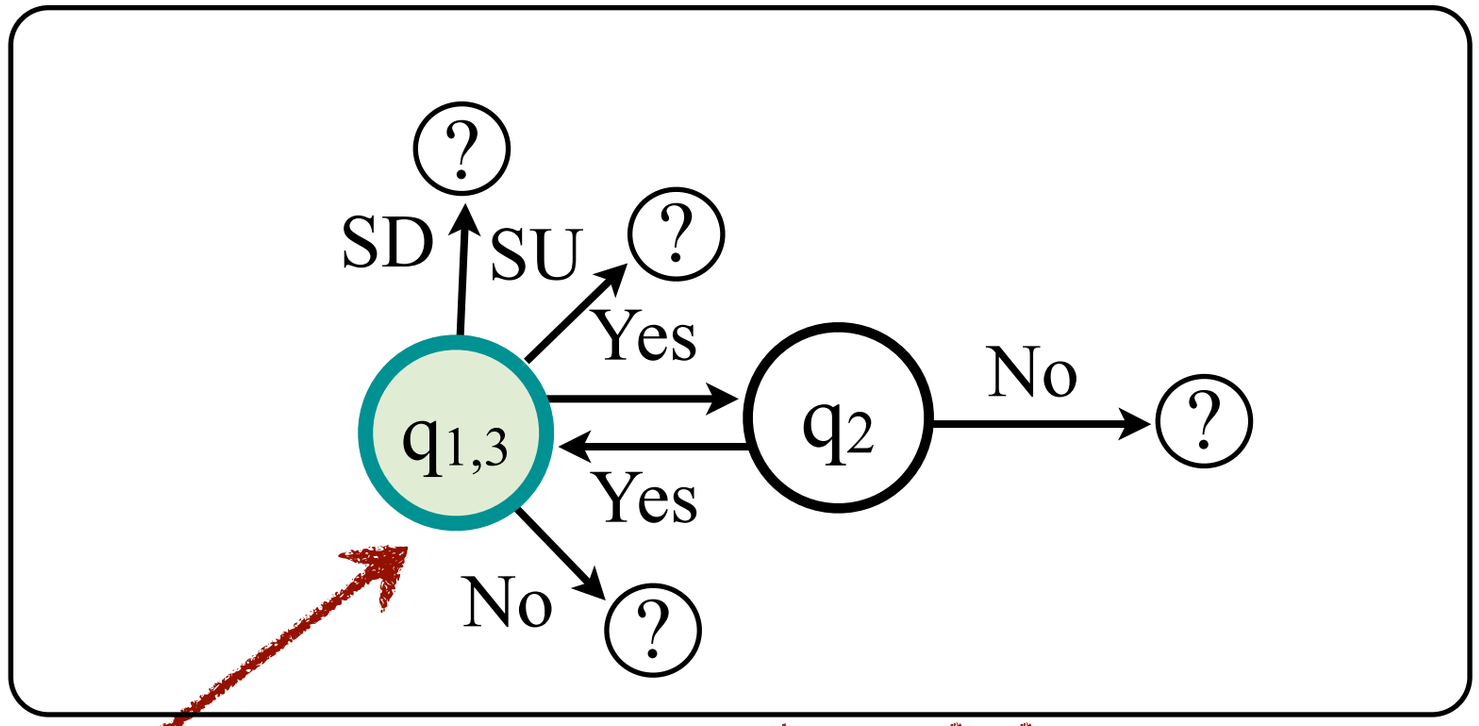
SwiftHand with Example

Iteration 2

Actual Application



Model Learned



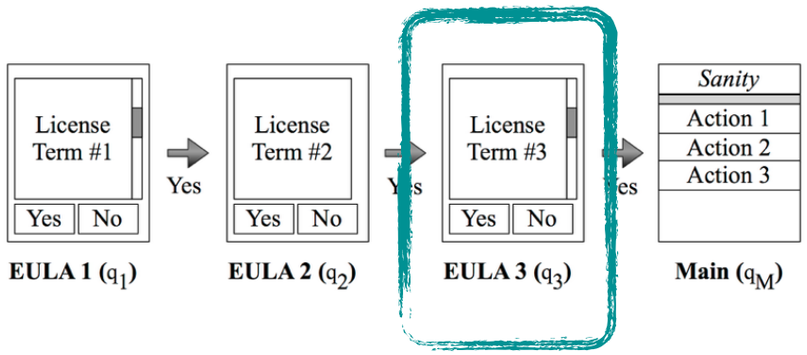
Optimistic merging

Expectation : a new screen will lead to interesting states

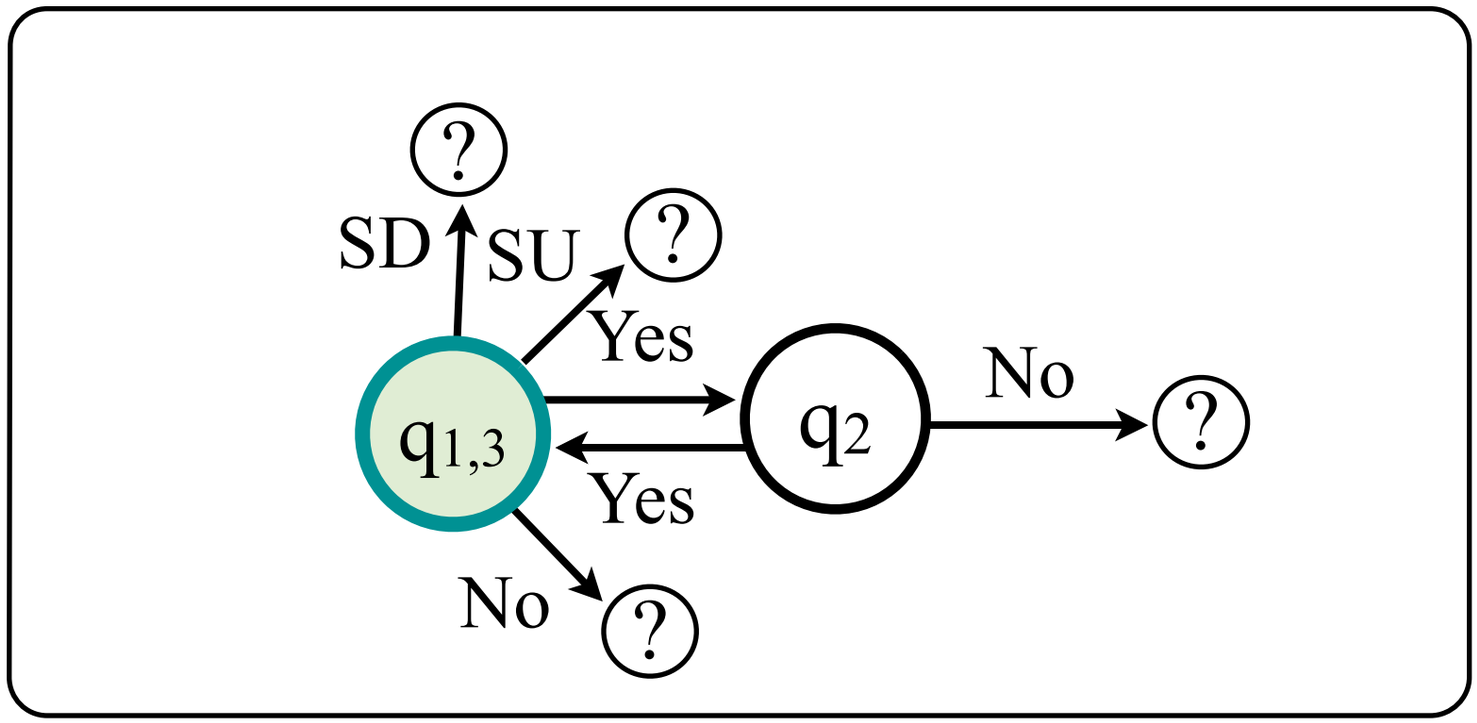
SwiftHand with Example

Iteration 3

Actual Application



Model Learned

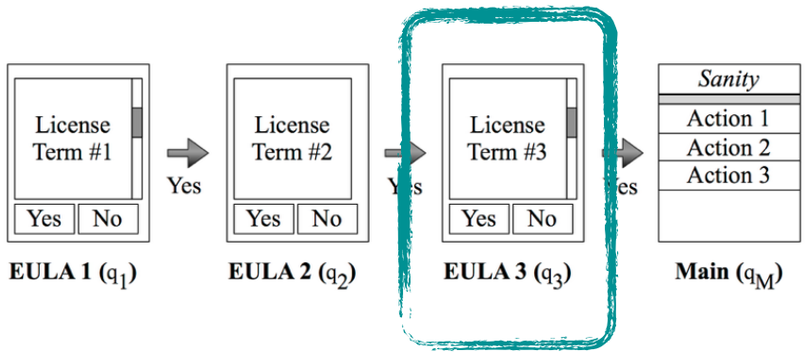


pick a target

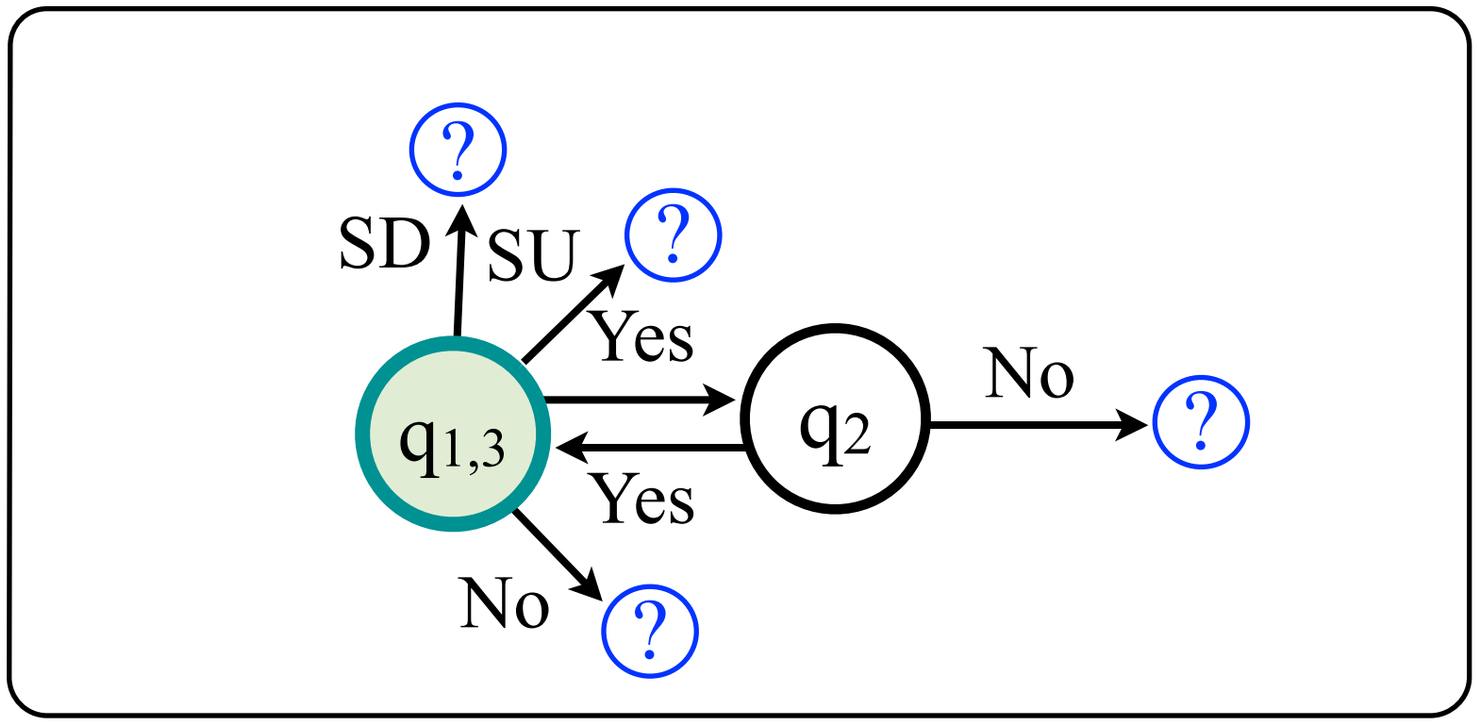
SwiftHand with Example

Iteration 3

Actual Application



Model Learned

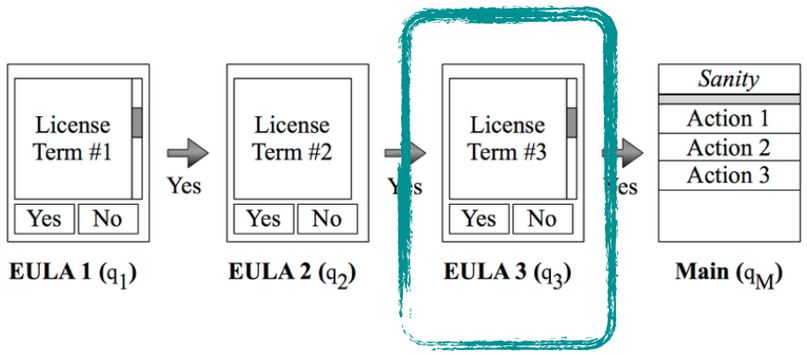


pick a target

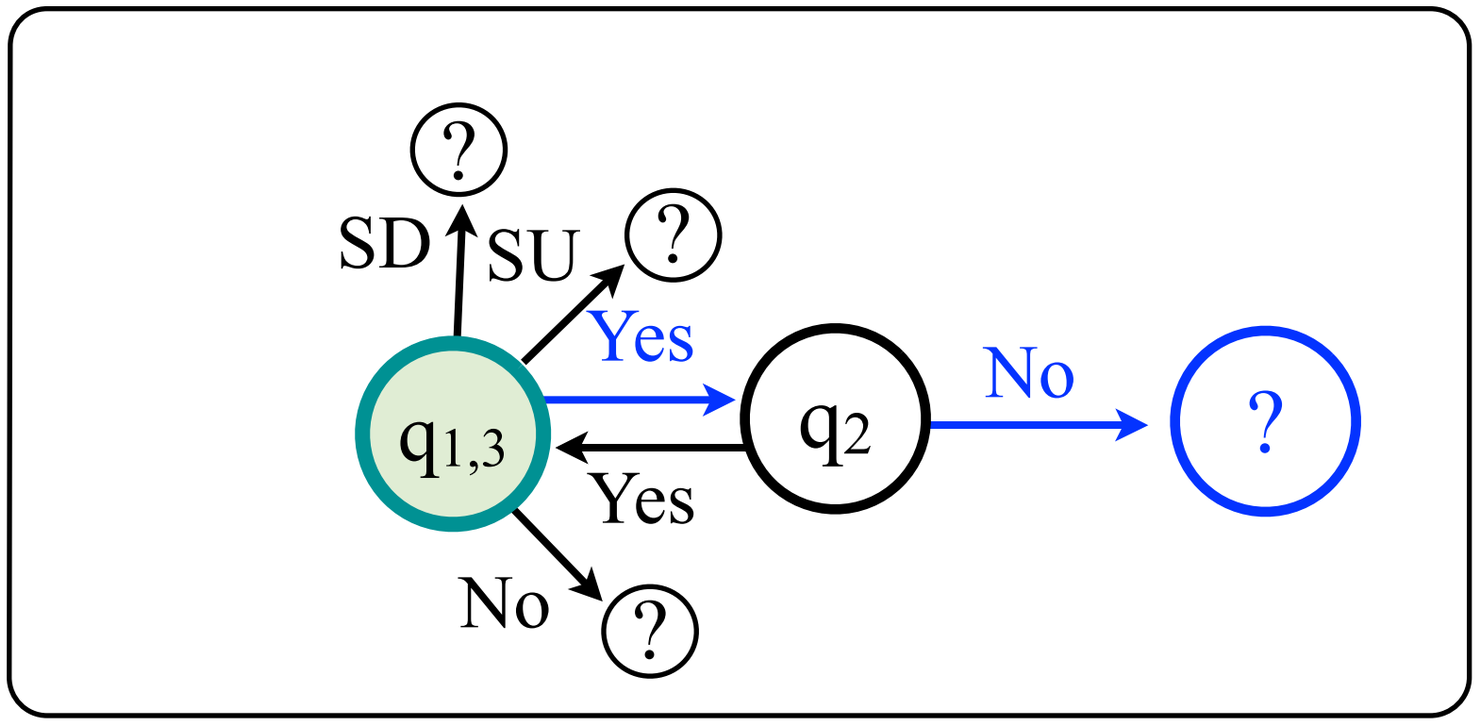
SwiftHand with Example

Iteration 3

Actual Application



Model Learned

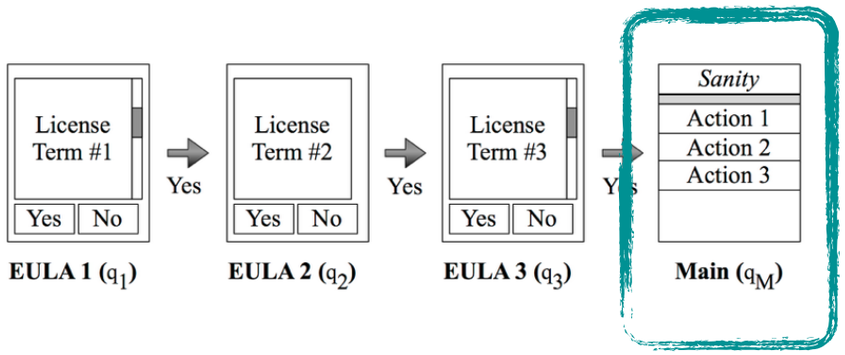


pick a target

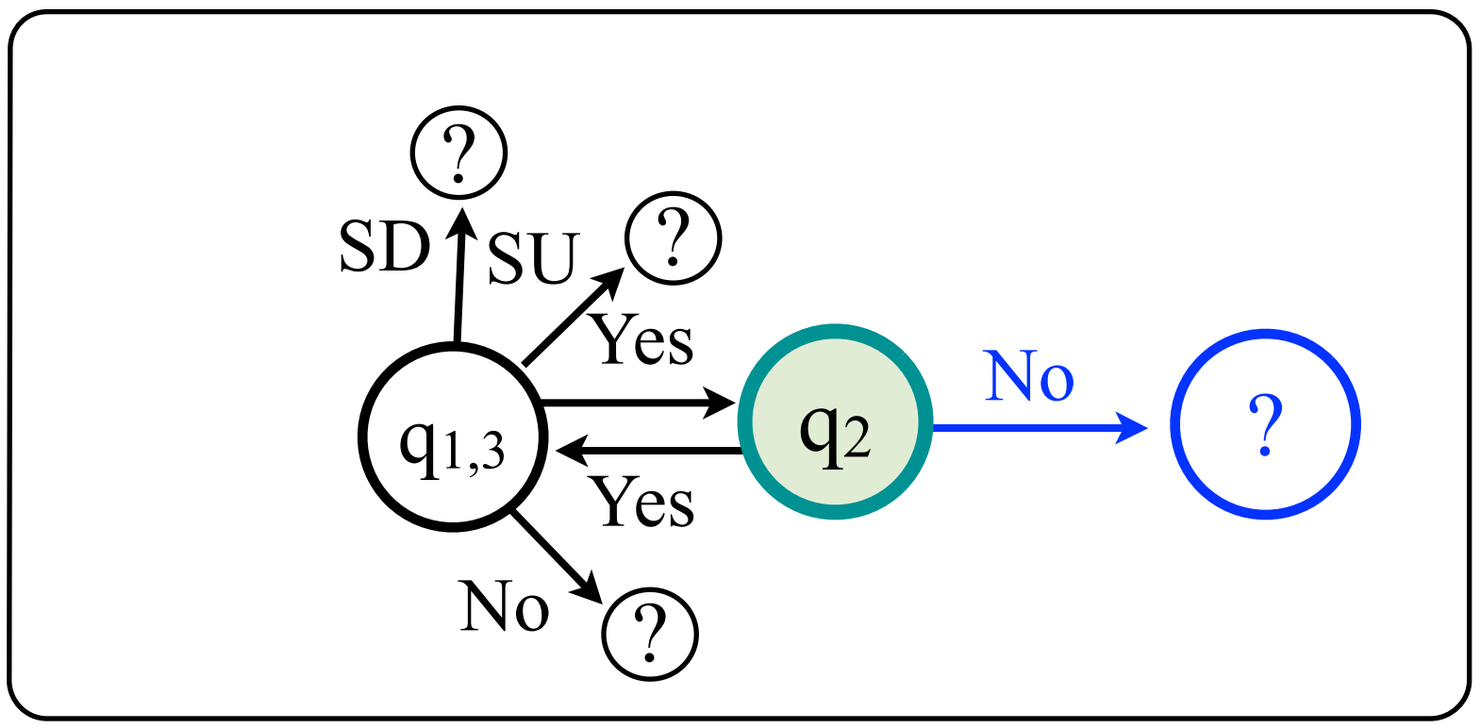
SwiftHand with Example

Iteration 3

Actual Application



Model Learned

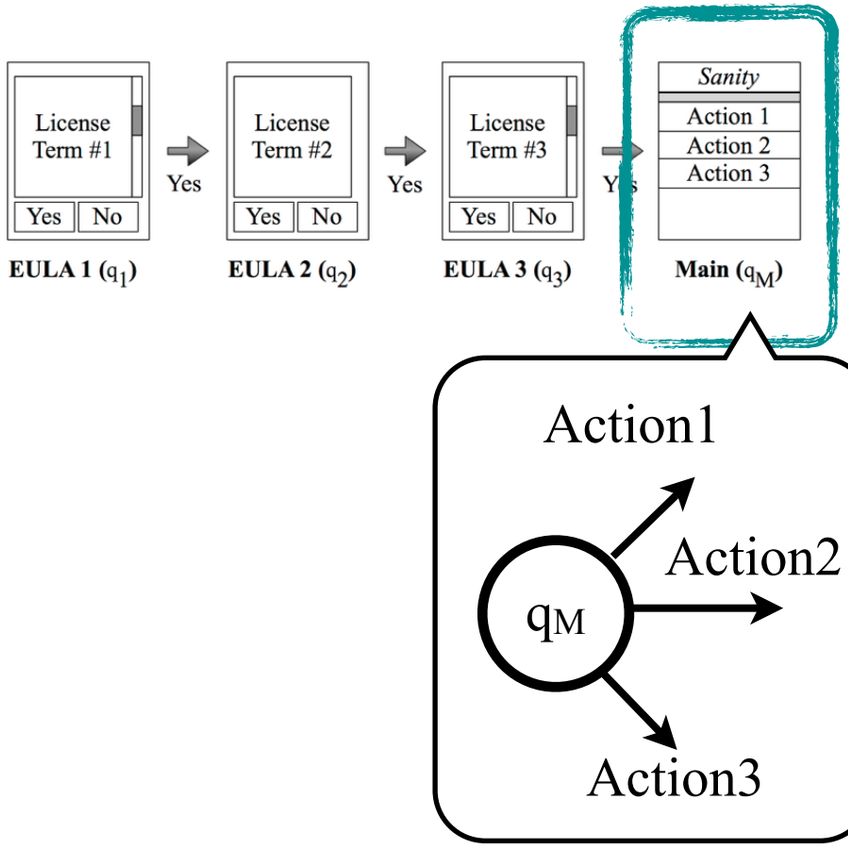


execute

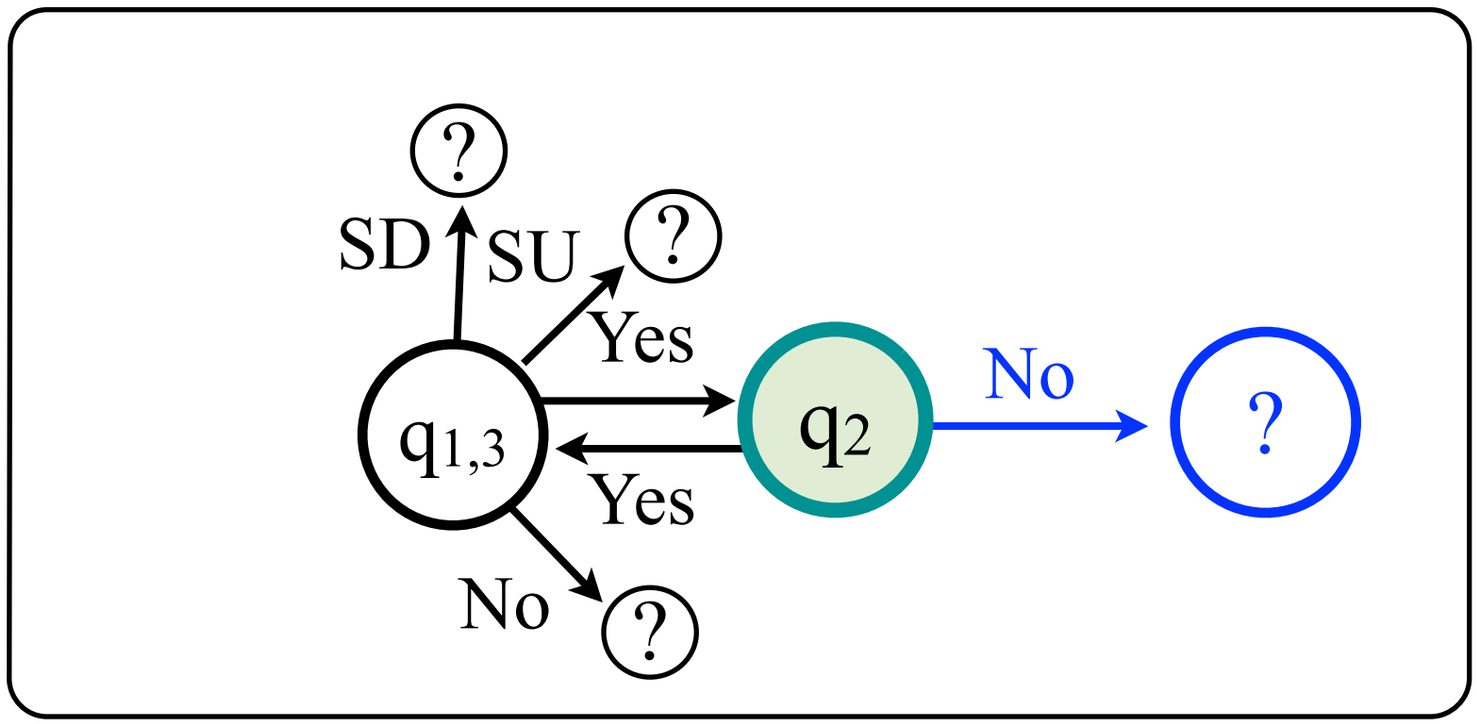
SwiftHand with Example

Iteration 3

Actual Application



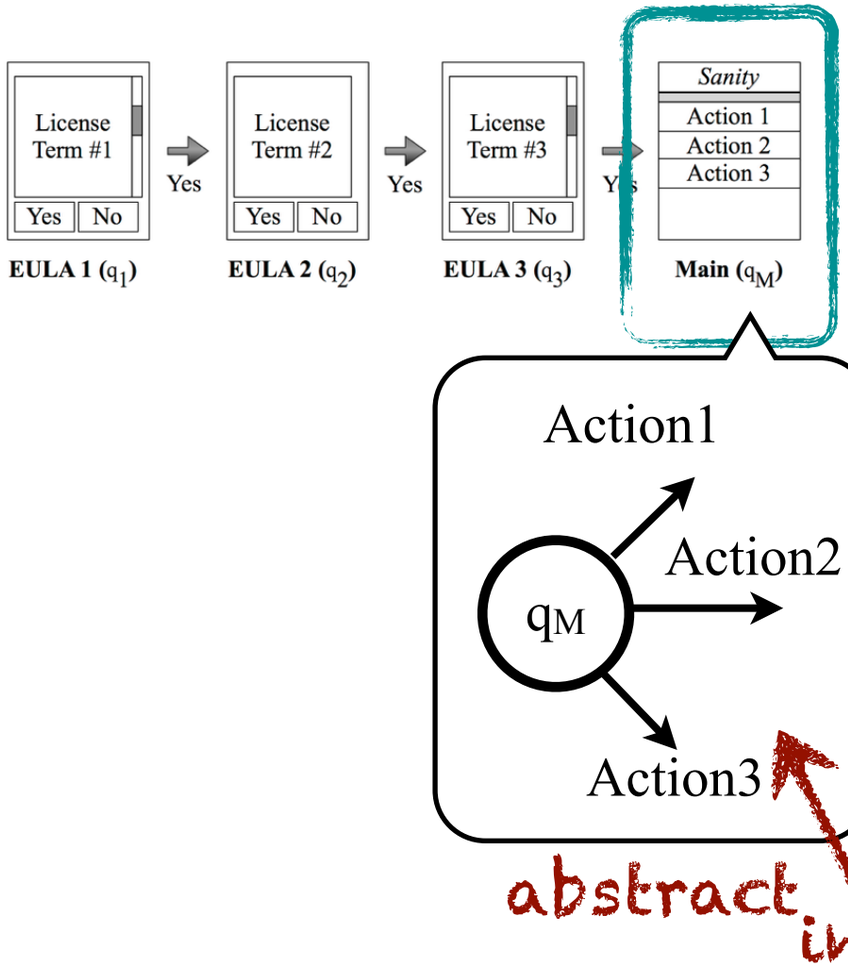
Model Learned



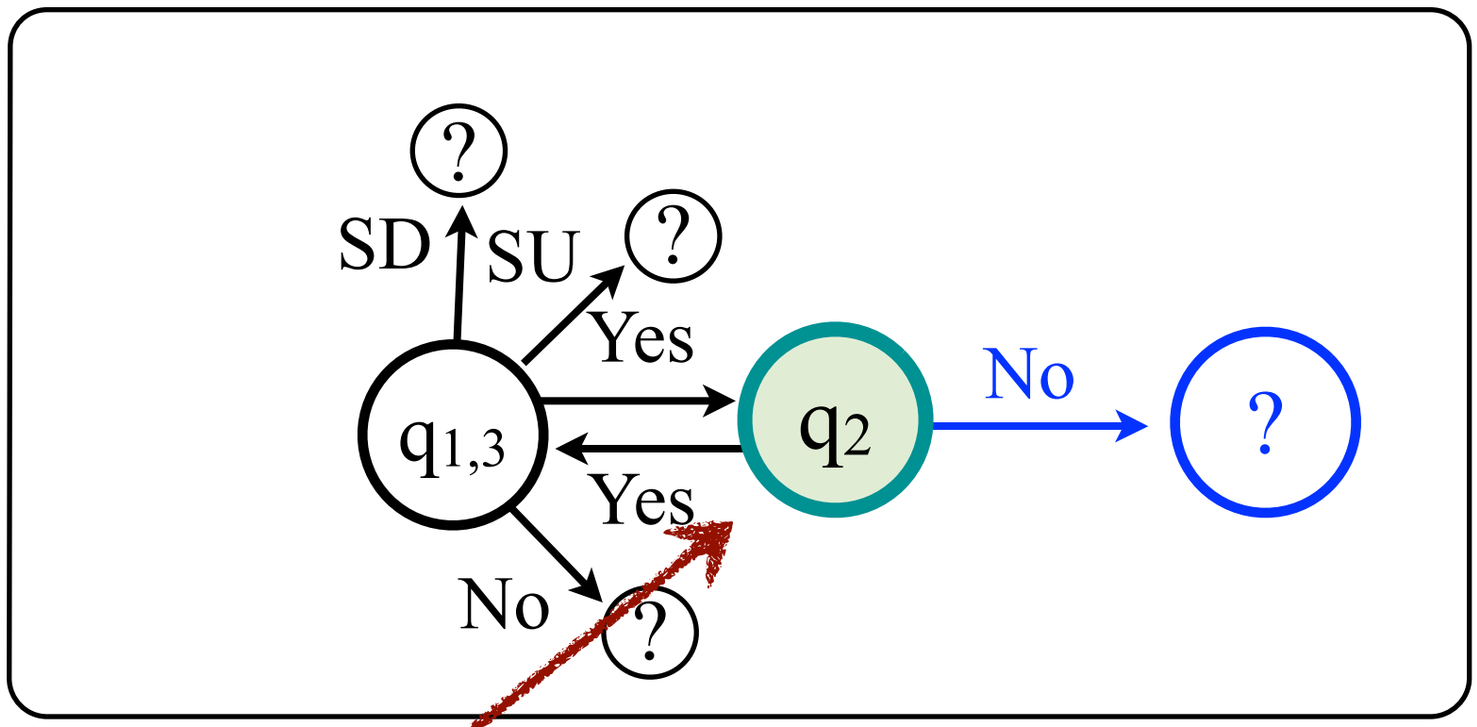
SwiftHand with Example

Iteration 3

Actual Application



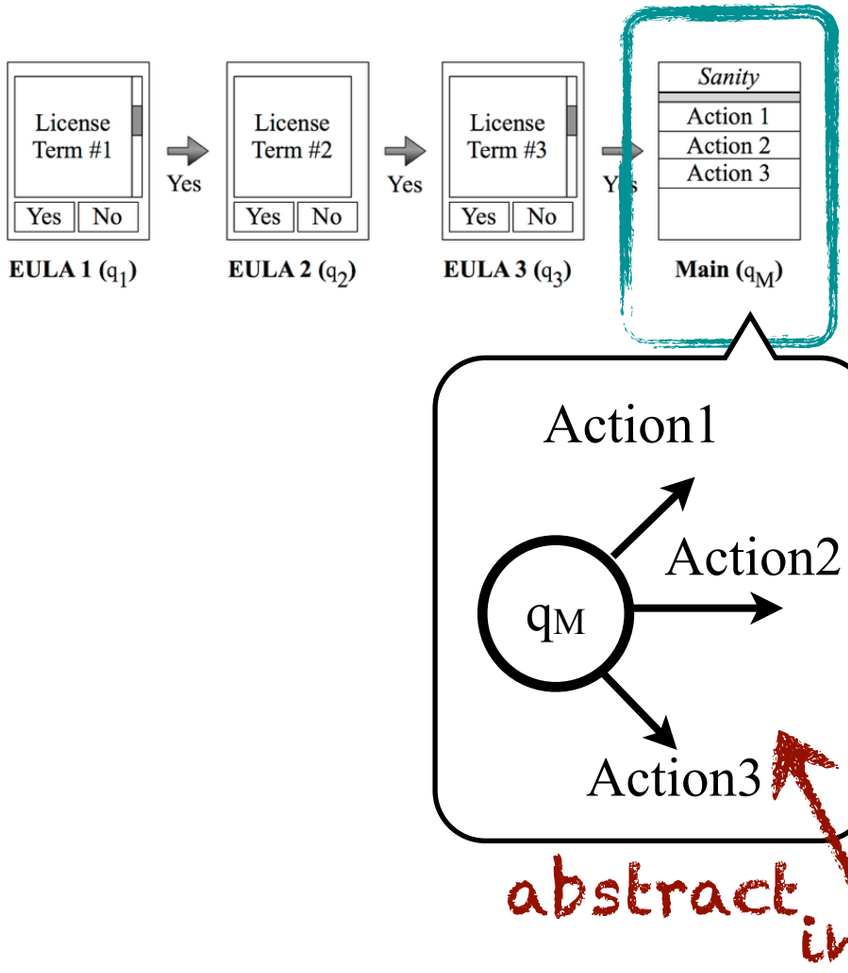
Model Learned



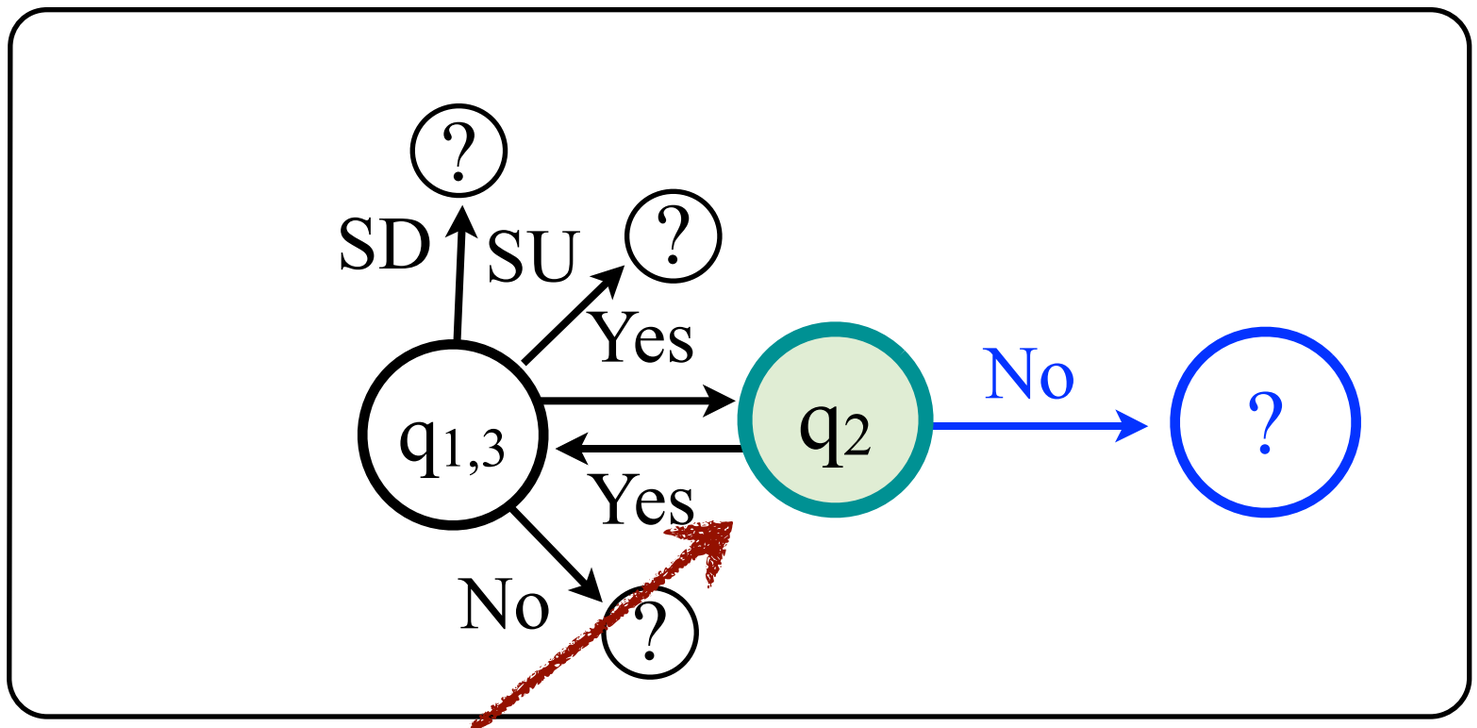
SwiftHand with Example

Iteration 3

Actual Application



Model Learned

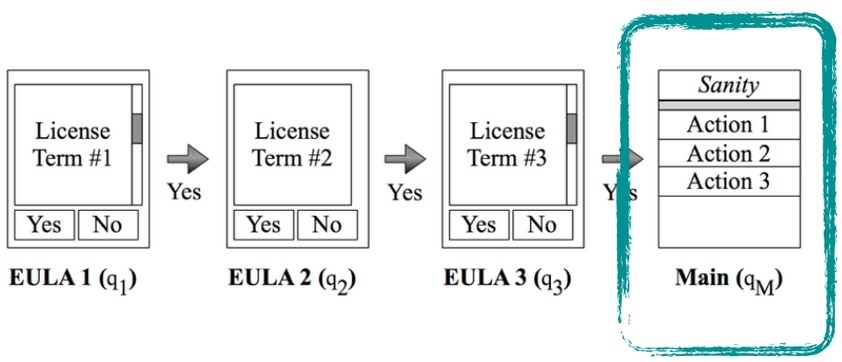


Some of previous merges were to aggressive

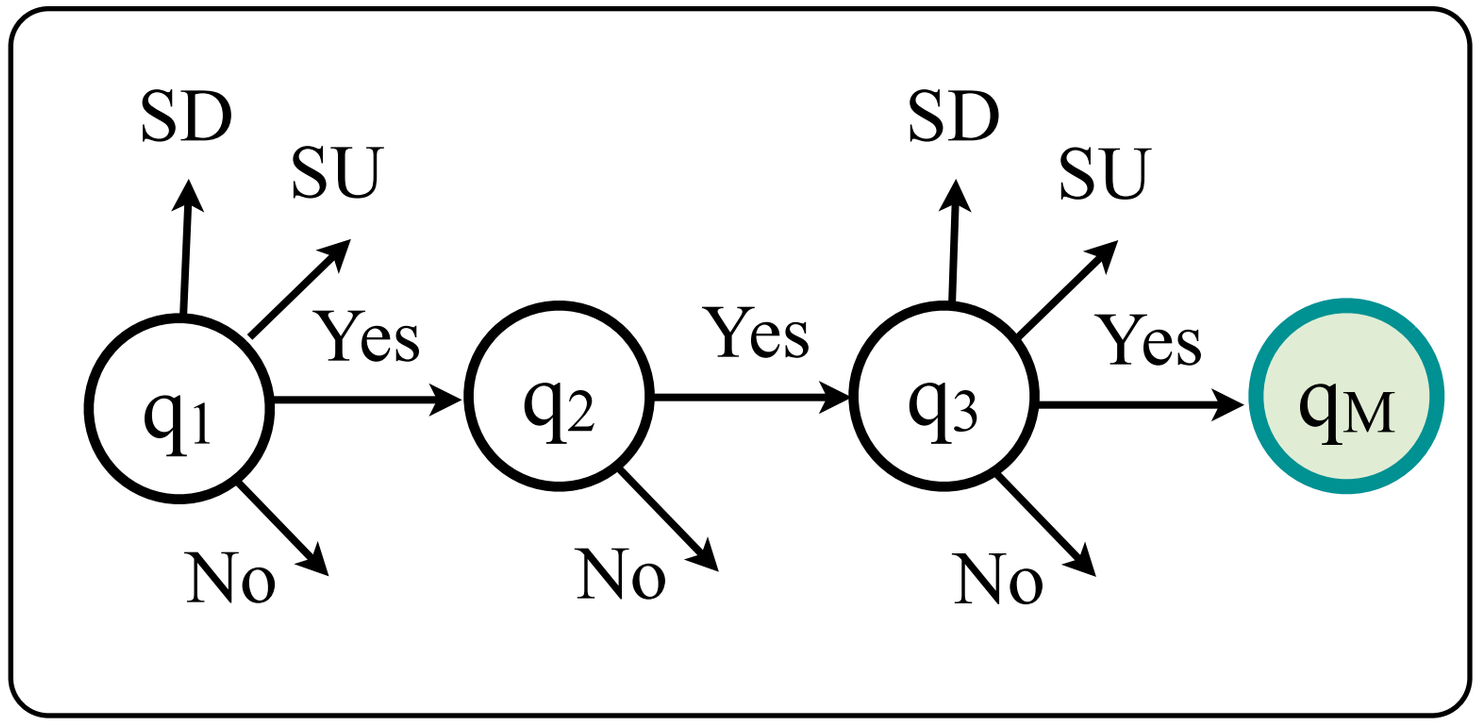
SwiftHand with Example

Iteration 3

Actual Application



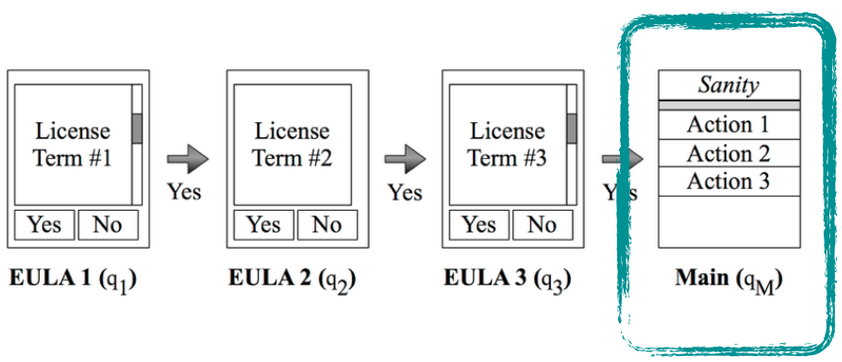
Model Learned



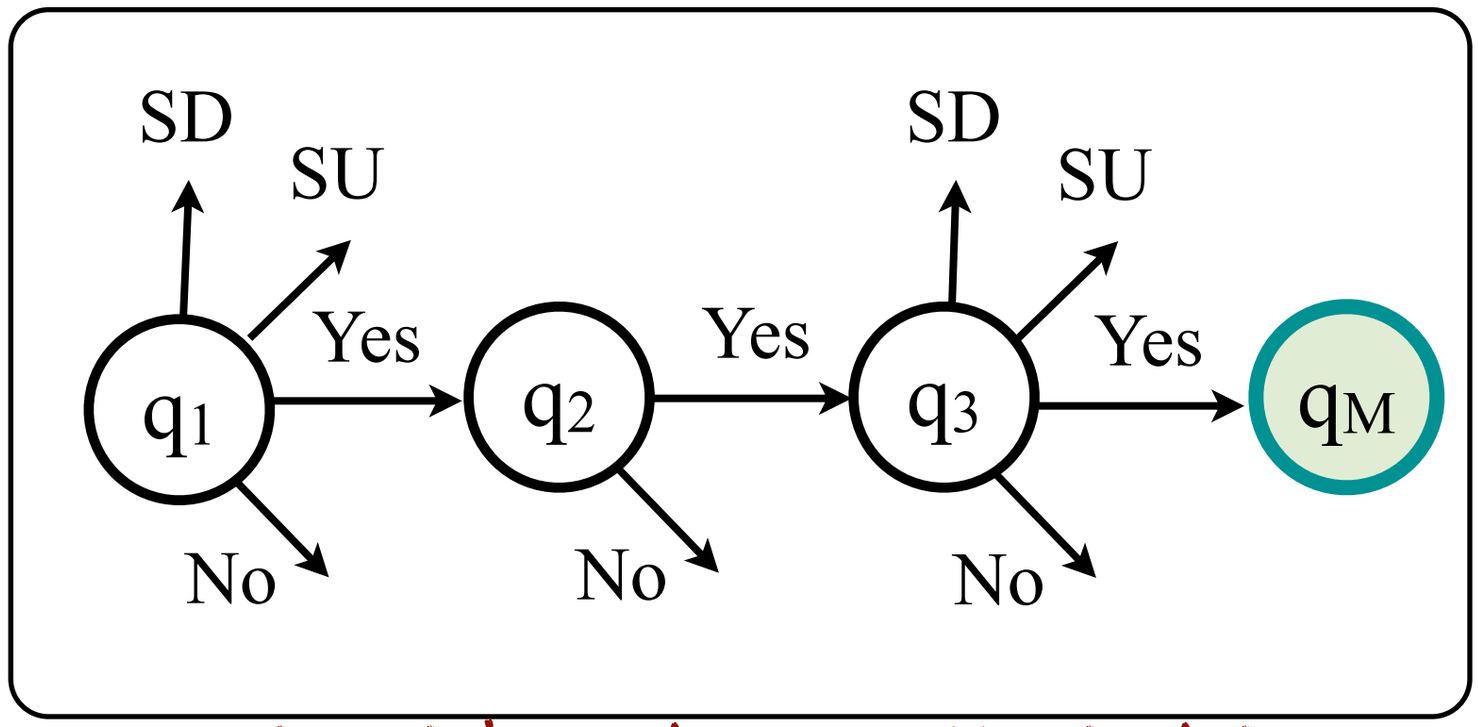
SwiftHand with Example

Iteration 3

Actual Application



Model Learned

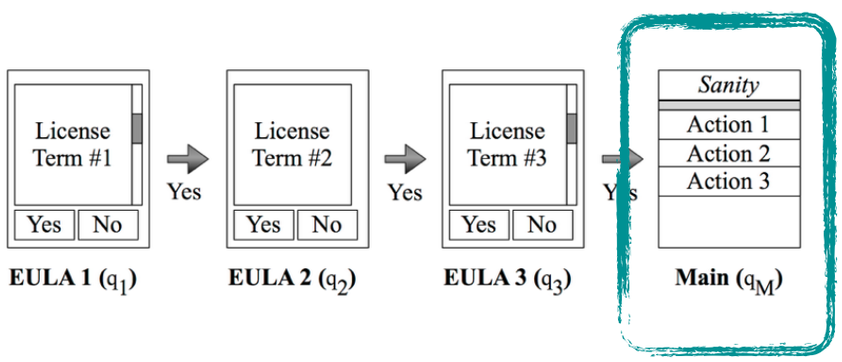


reconstruct based on collected traces

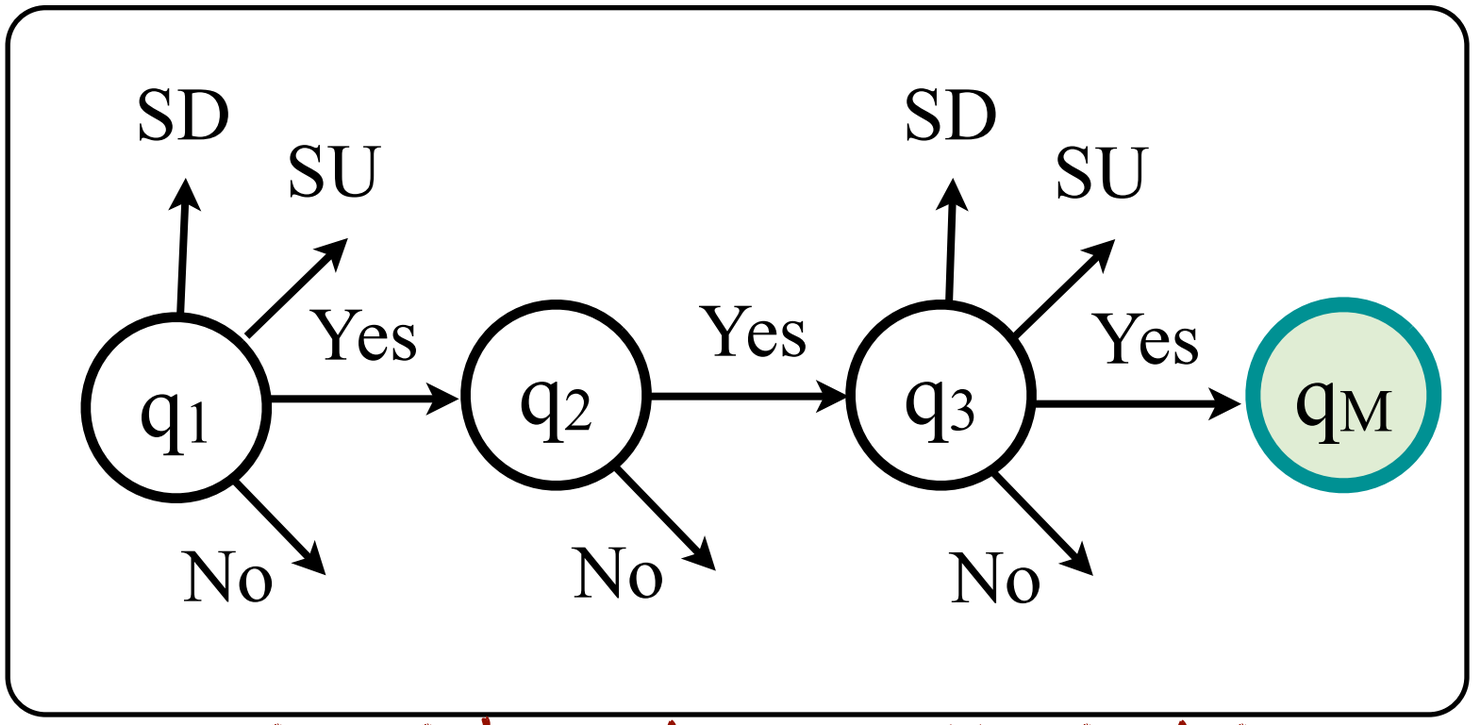
SwiftHand with Example

Iteration 3

Actual Application



Model Learned



reconstruct based on collected traces

Reconstruction uses off-line learning algorithm

(*without re-executing the program)

SwiftHand

Summary

- Combining learning and GUI testing
  - FSM model (state = screen / transition = event)
  - **Approximate learning** with optimistic state merging
    - helps to explore interesting states quickly
  - **Avoids restarts**
    - helps to reduce testing time
  - Perform passive learning
    - helps to keep model consistent with the app

Experiments

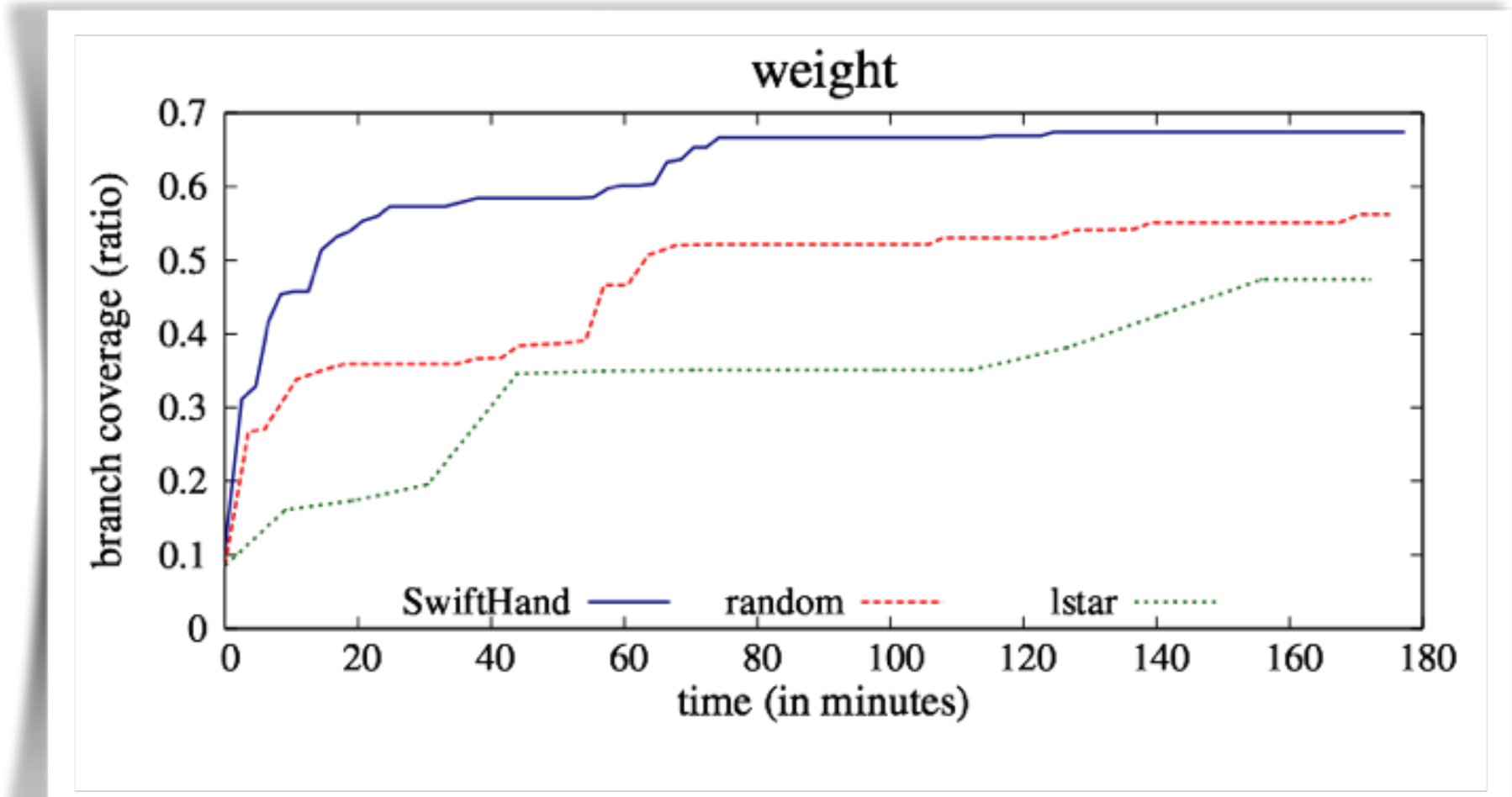
Setup

- *SwiftHand* vs **Random** vs *L**
- 10 Android apps (F-droid app market)
- 3 hours of testing per strategy
- On an Android emulator

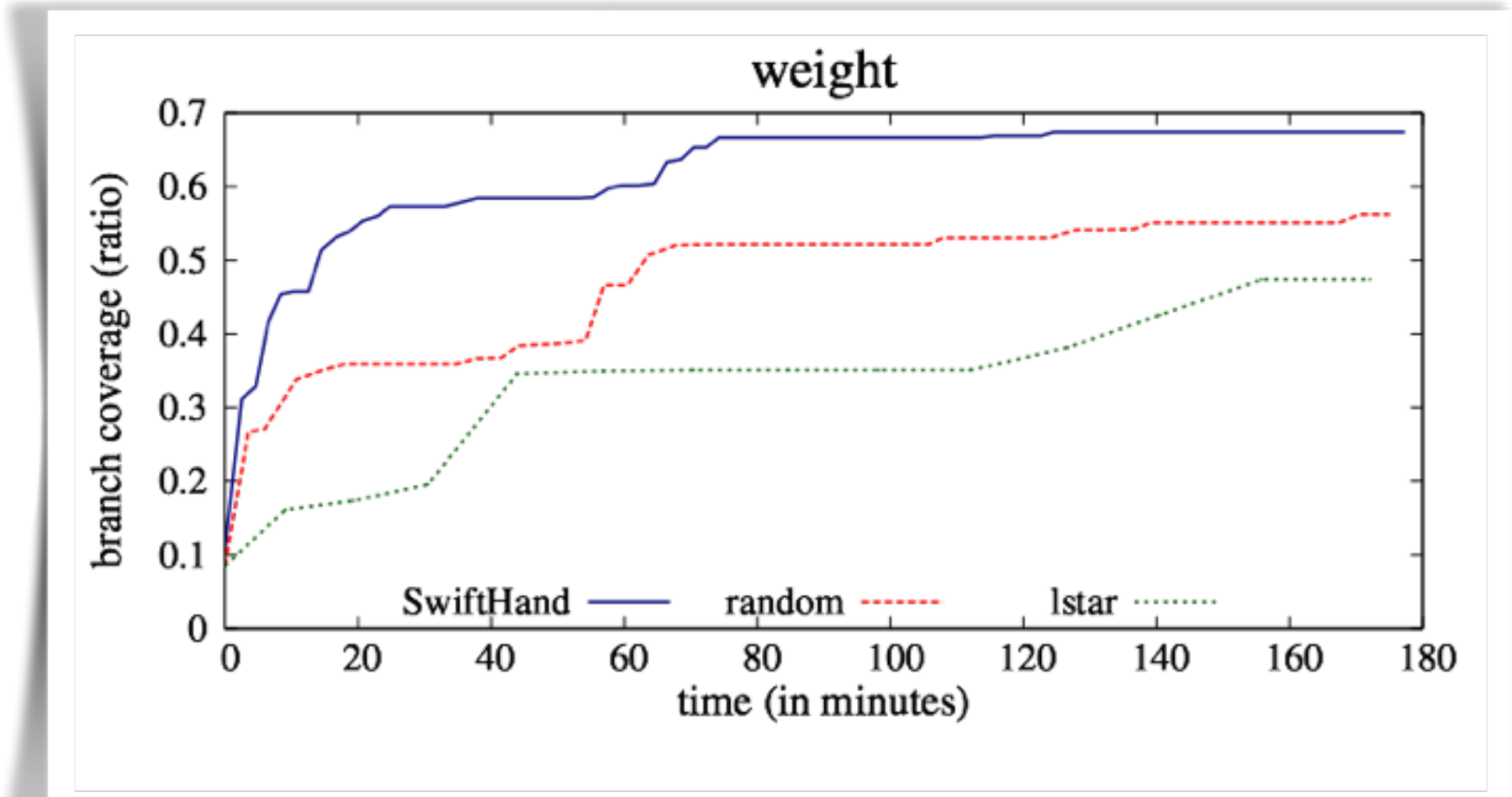
Measurement

- Branch coverage (final coverage, coverage speed)

Experiments - Coverage

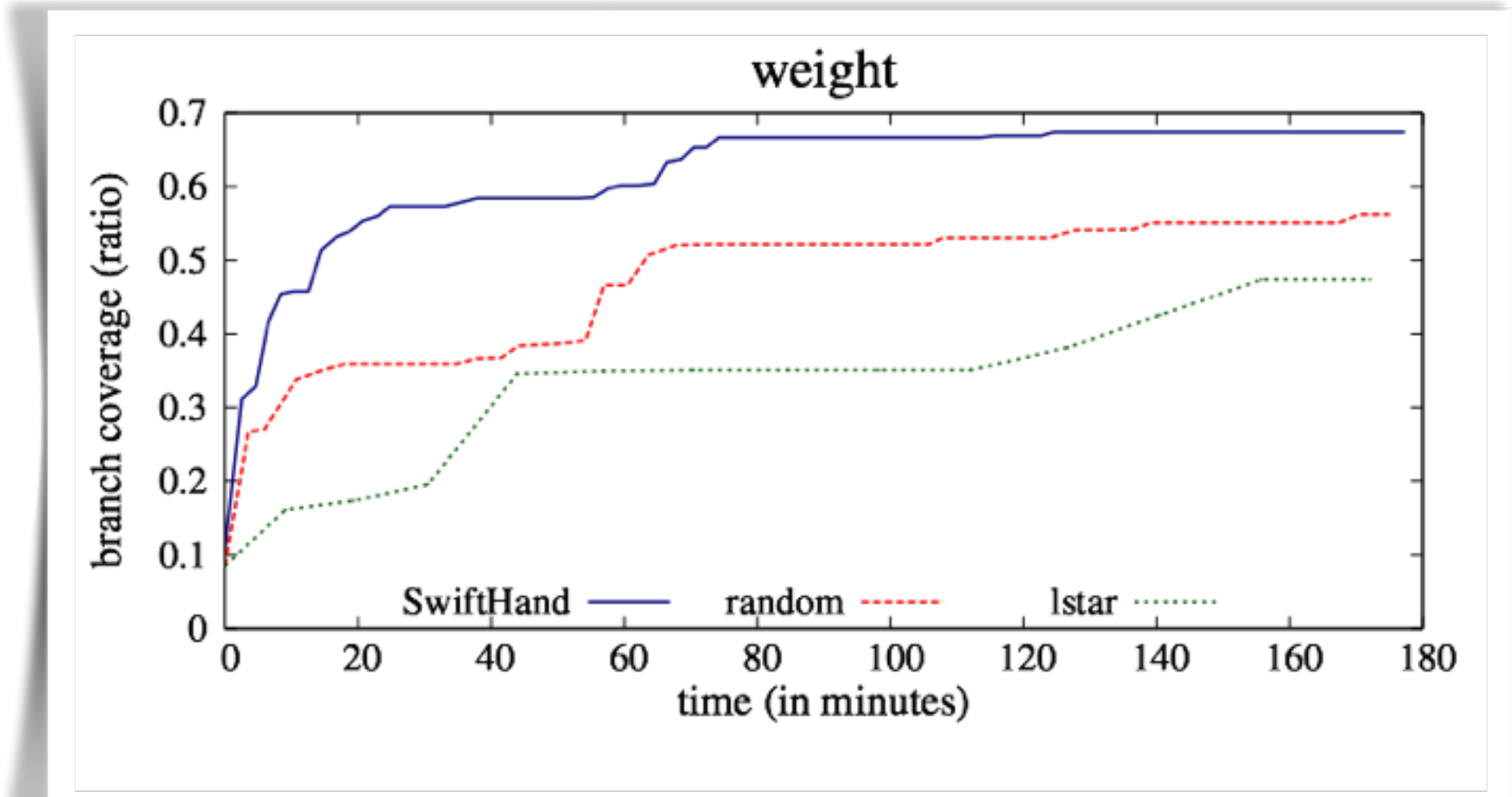


Experiments - Coverage



- *SwiftHand* has the fastest coverage speed
- *SwiftHand* has the highest final coverage

Experiments - Coverage

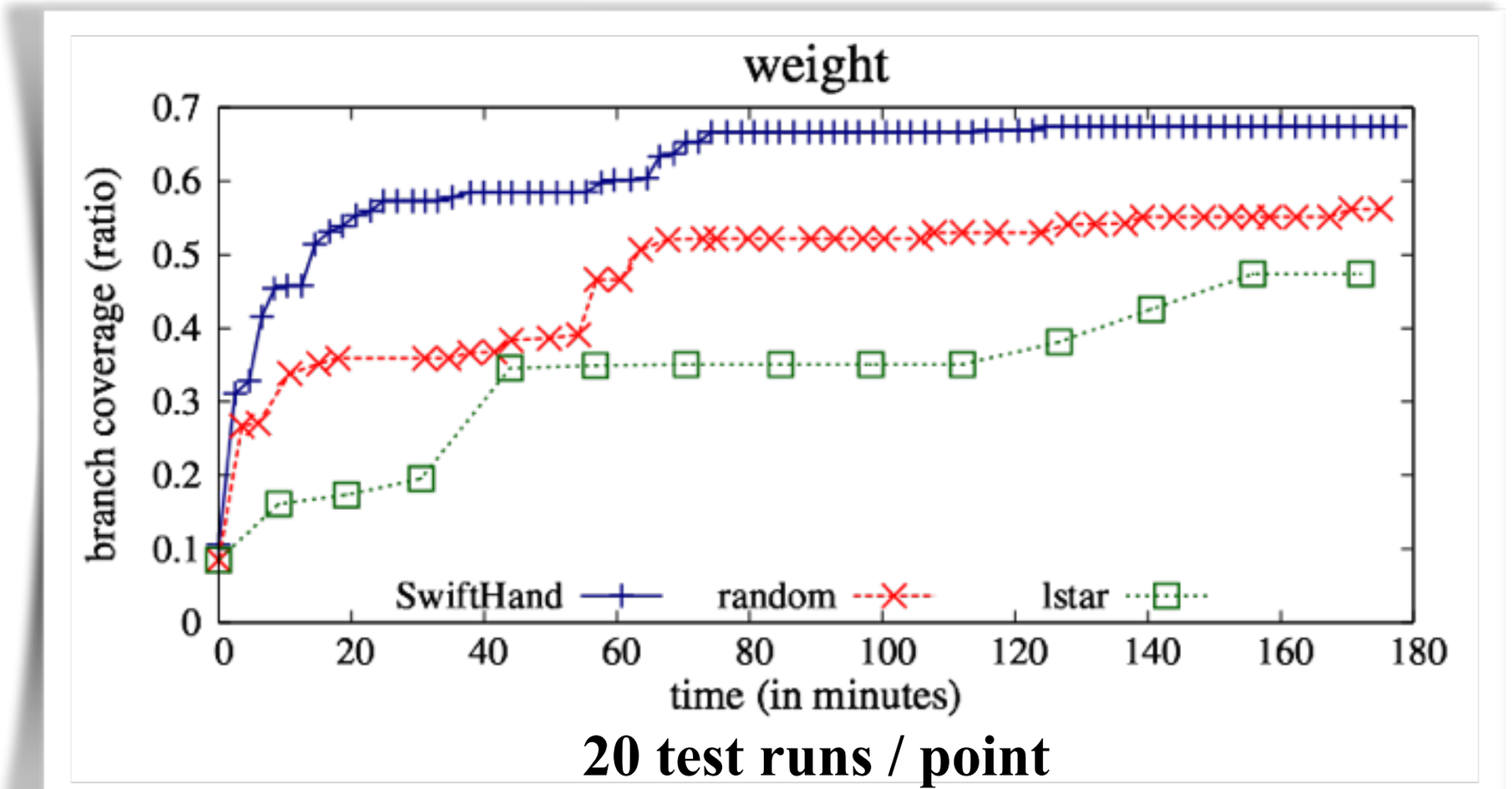


- *SwiftHand* has the fastest coverage speed
- *SwiftHand* has the highest final coverage

SwiftHand >> **Random** >> *L**

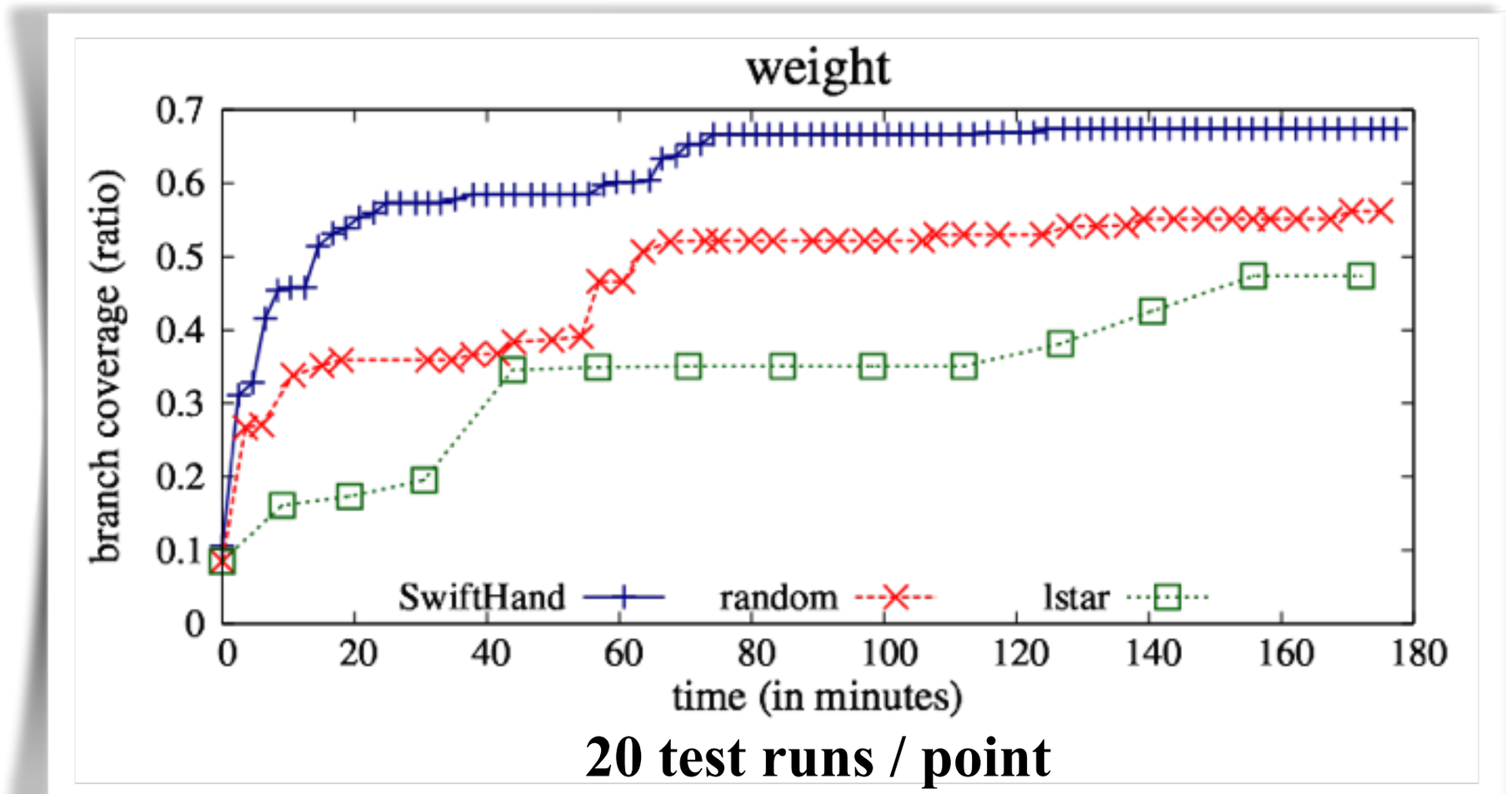
(for all 10 apps)

Experiments - Cost of Test



20 test runs / point

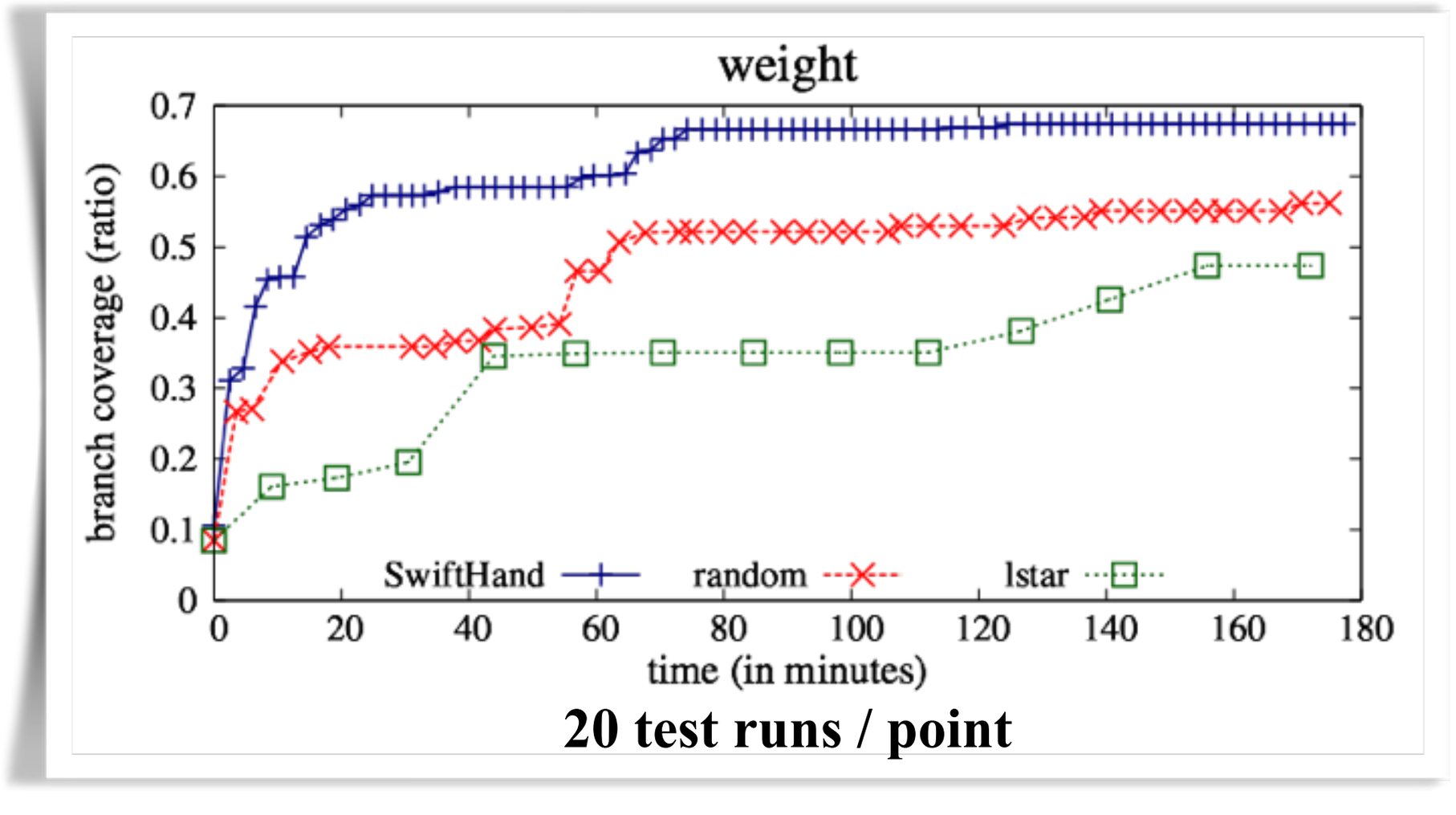
Experiments - Cost of Test



Time comparison:

one *L** run = 7 *SwiftHand* runs = 4 **Random** runs

Experiments - Cost of Test



20 test runs / point

Time comparison:

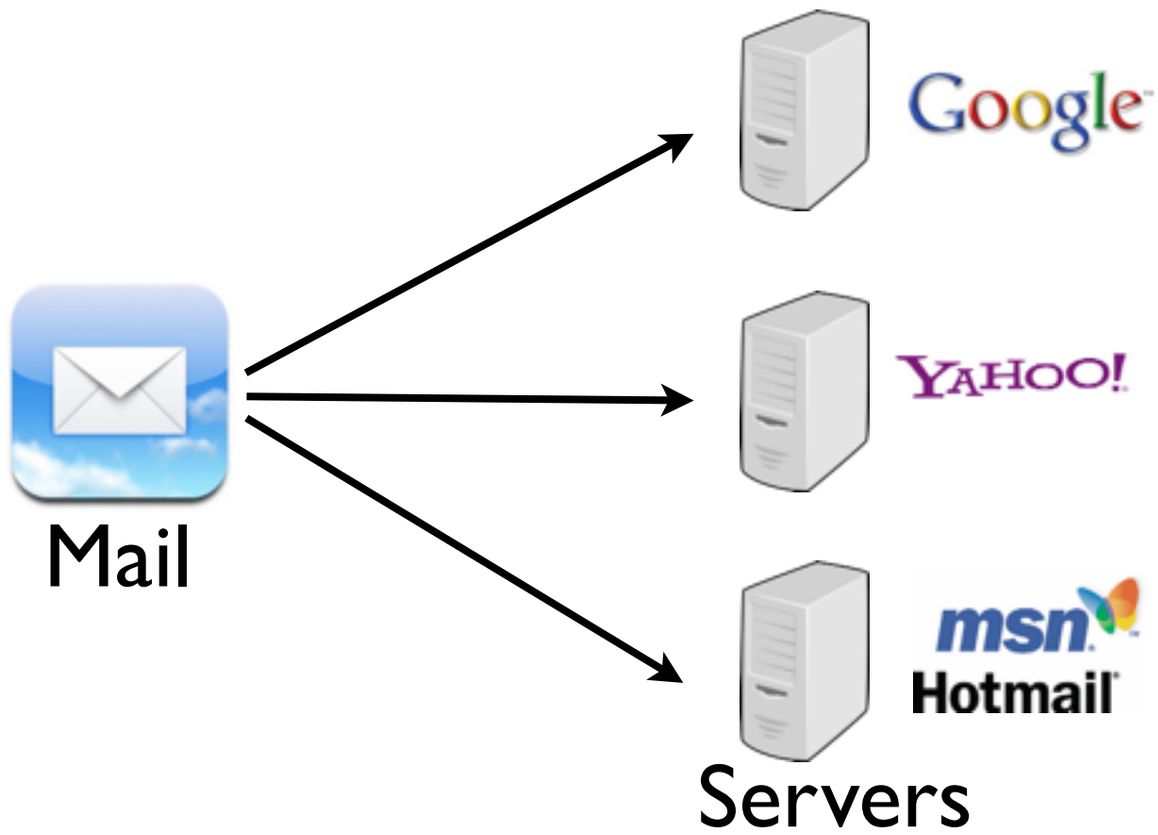
one *L** run = 7 *SwiftHand* runs = 4 **Random** runs

*L** >> **Random** >> **SwiftHand**

(for all 10 apps)

Challenges and Future Work

- External states

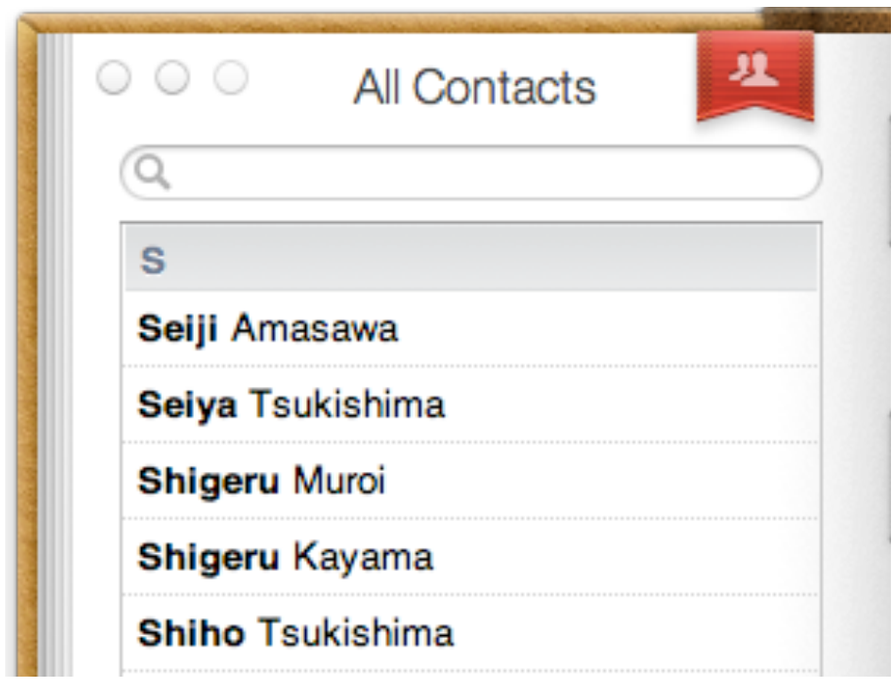


App state = Local state + External state

How to clean restart an app?

Challenges and Future Work

- External states
- String inputs



Application behavior depends on strings
How to pick strings well?

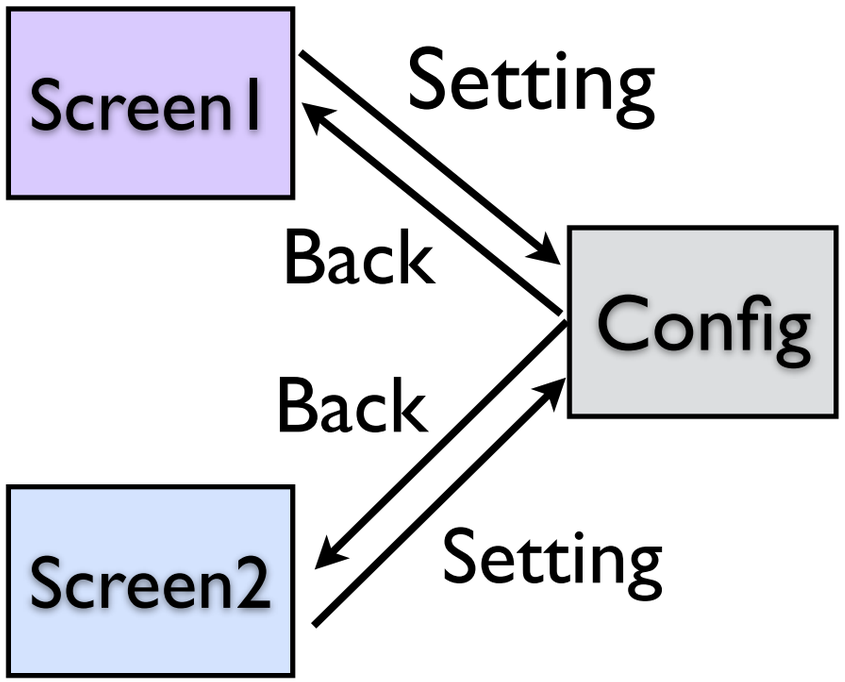
Challenges and Future Work

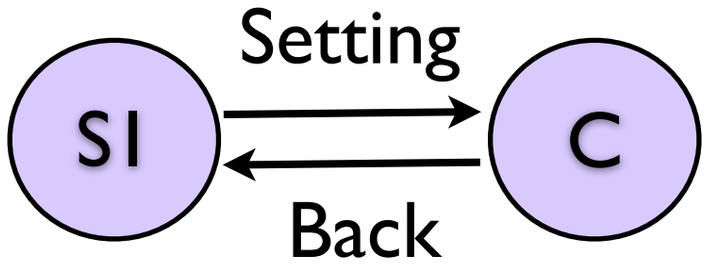
- External states
- String inputs
- Combinatorial explosion

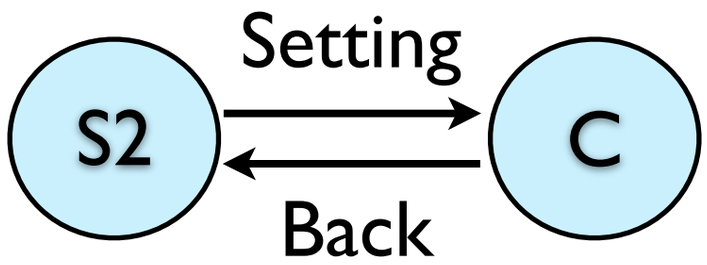
Component local states are multiplied in the model.
How to shrink search space?

Challenges and Future Work

- External states
- String inputs
- Combinatorial explosion



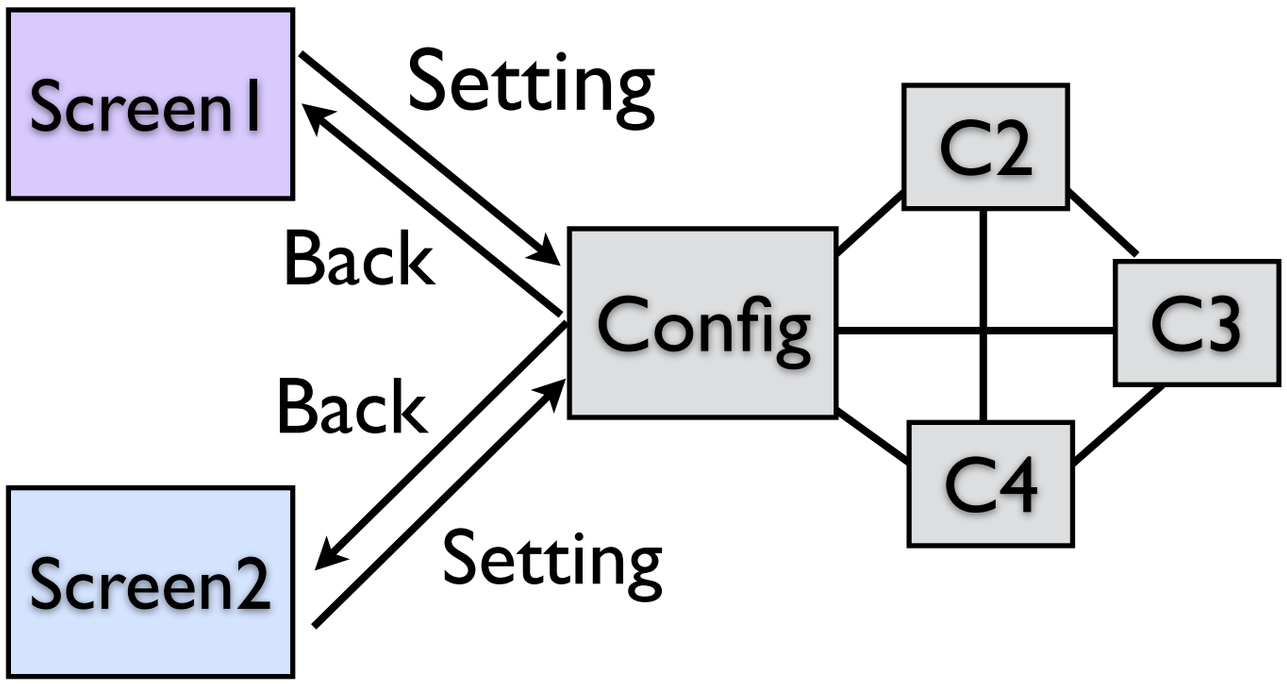


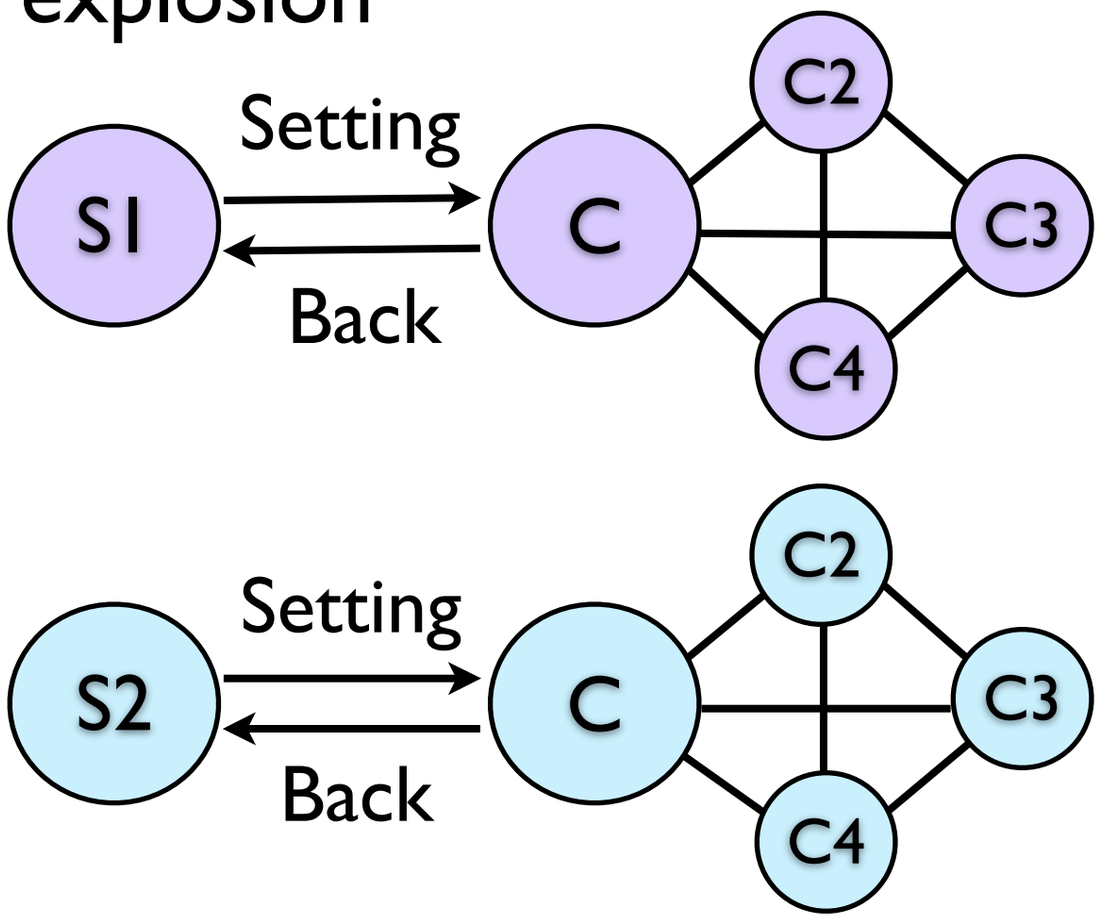


Component local states are multiplied in the model.
How to shrink search space?

Challenges and Future Work

- External states
- String inputs
- Combinatorial explosion





Component local states are multiplied in the model.
How to shrink search space?

Related Works

Automated model based GUI testing

- GUITAR
  - *B.Nguyen, B.Robbins, I.Benerjee, and A.Memo,* ASE 2013
  - *Atiff Memon,* STVR 2007

- CrawlJax
  - *A.Mesbah, A.v.Deursen, and S.Lenselink, TWEB 2012*
  - *A.Mesbah, A.v.Deursen, and S.Lenselink, ICWE 2008*

Testing with model learning

- Survey papers
  - *K.Meinke, F.Niu, and M.Sindhu, ISoLa 2011*
  - *D.Lee and M.Yannakakis, IEEE 1996*

Conclusion

Guided GUI Testing of Android Apps with Minimal Restart and Approximate Learning

Conclusion

Guided GUI Testing of Android Apps with Minimal Restart and Approximate Learning

It worked.

Conclusion

Guided GUI Testing of Android Apps with Minimal Restart and Approximate Learning

Thank you!