Declarative representation and analysis of state machines

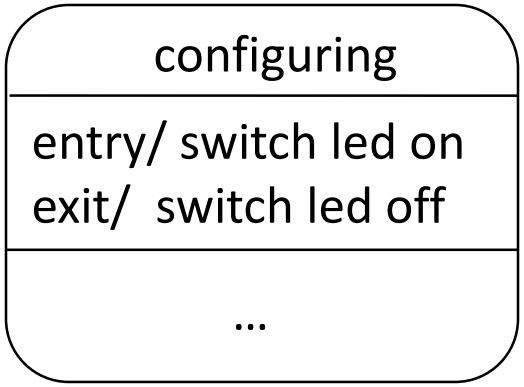
Dr. Constantinos Constantinides, P.Eng.
Computer Science and Software Engineering
Concordia University

Declarative representation of a UML State Machine

- A **UML State Machine** can be modeled as **a set of facts**:

Format	Description
`state/4:`	`state(Name, OnEntry, OnExit, Do).`
`initial/1`	`initial_state(Name).`
`final/1`	`final_state(Name).`
`transition/5:`	`transition(SourceState, TargetState, Event, Guard, Action).`

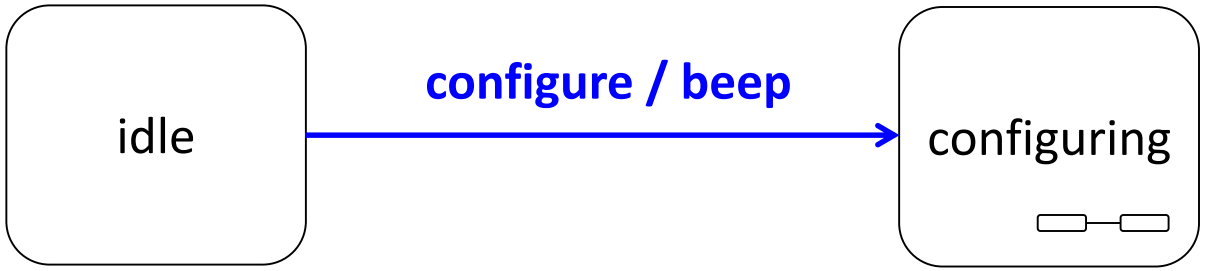
Capturing states



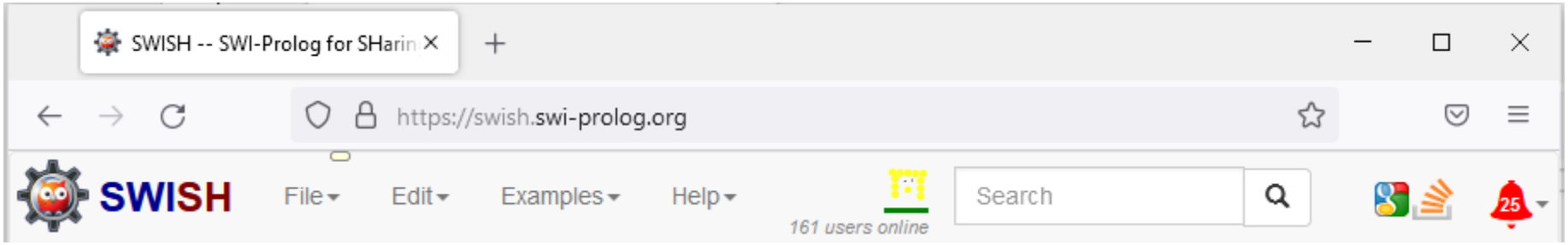

```
state(Name, OnEntry, OnExit, Do).
```

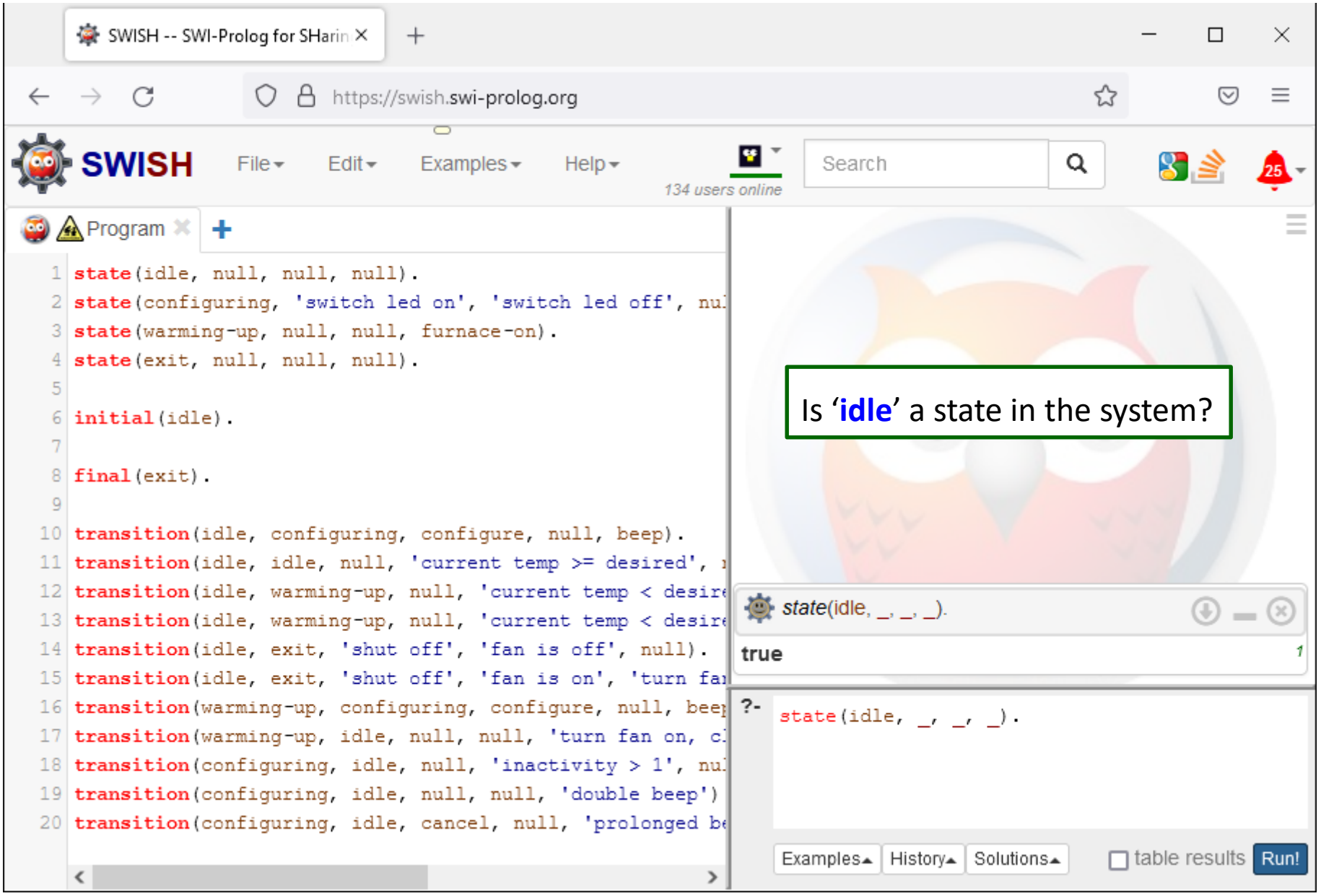
```
state(configuring, ‘switch led on’, ‘switch led off’, null).
```

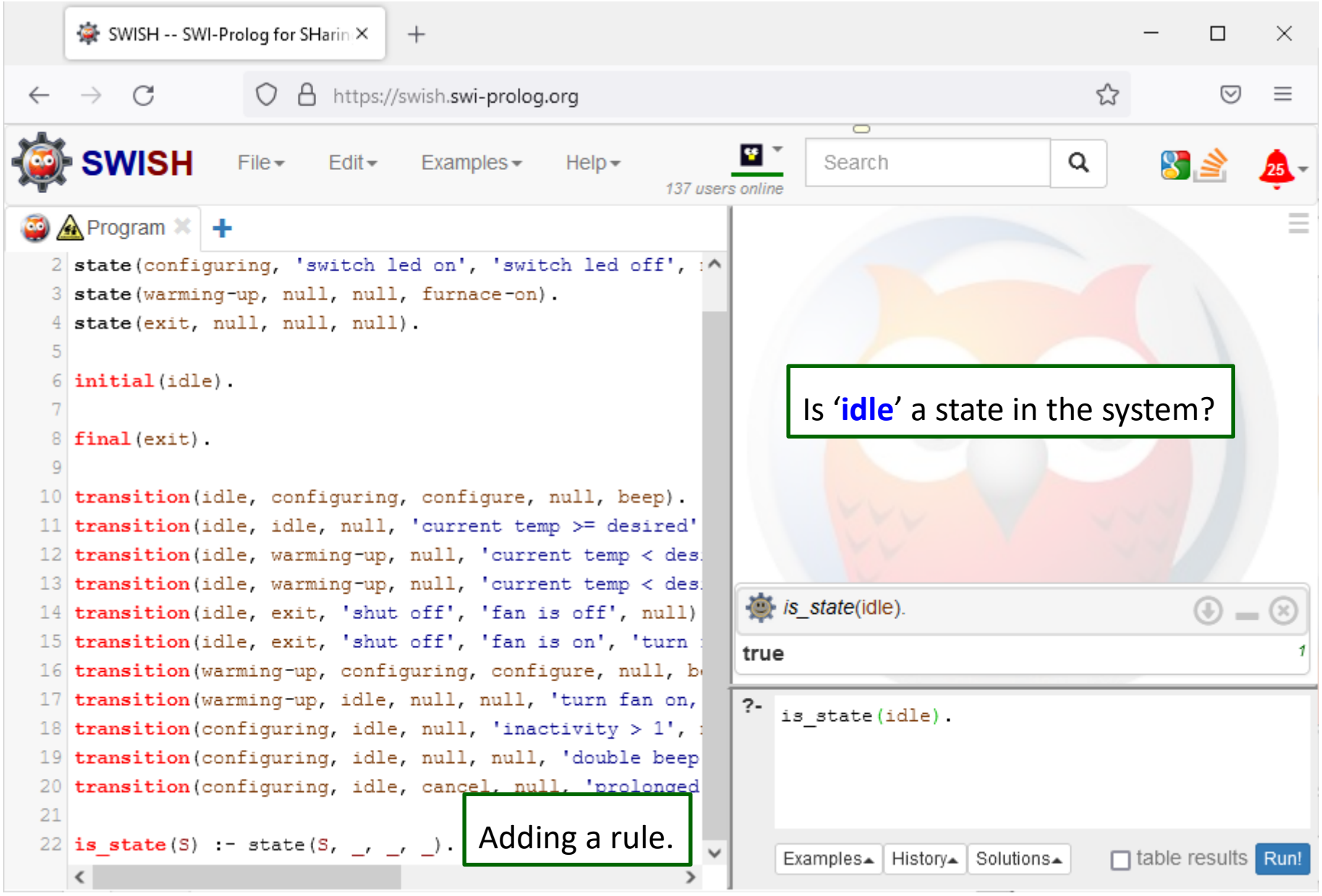
Capturing transitions

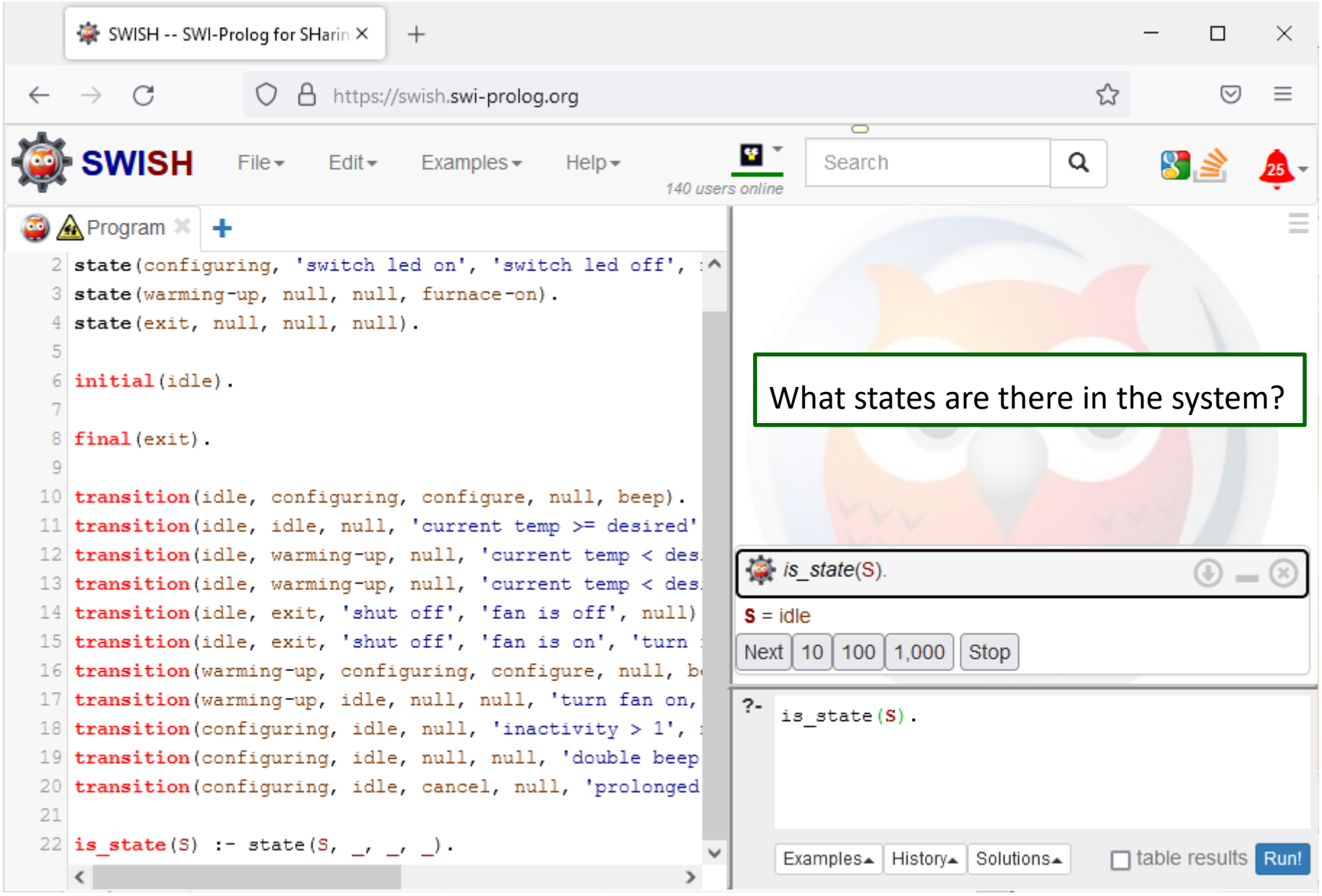


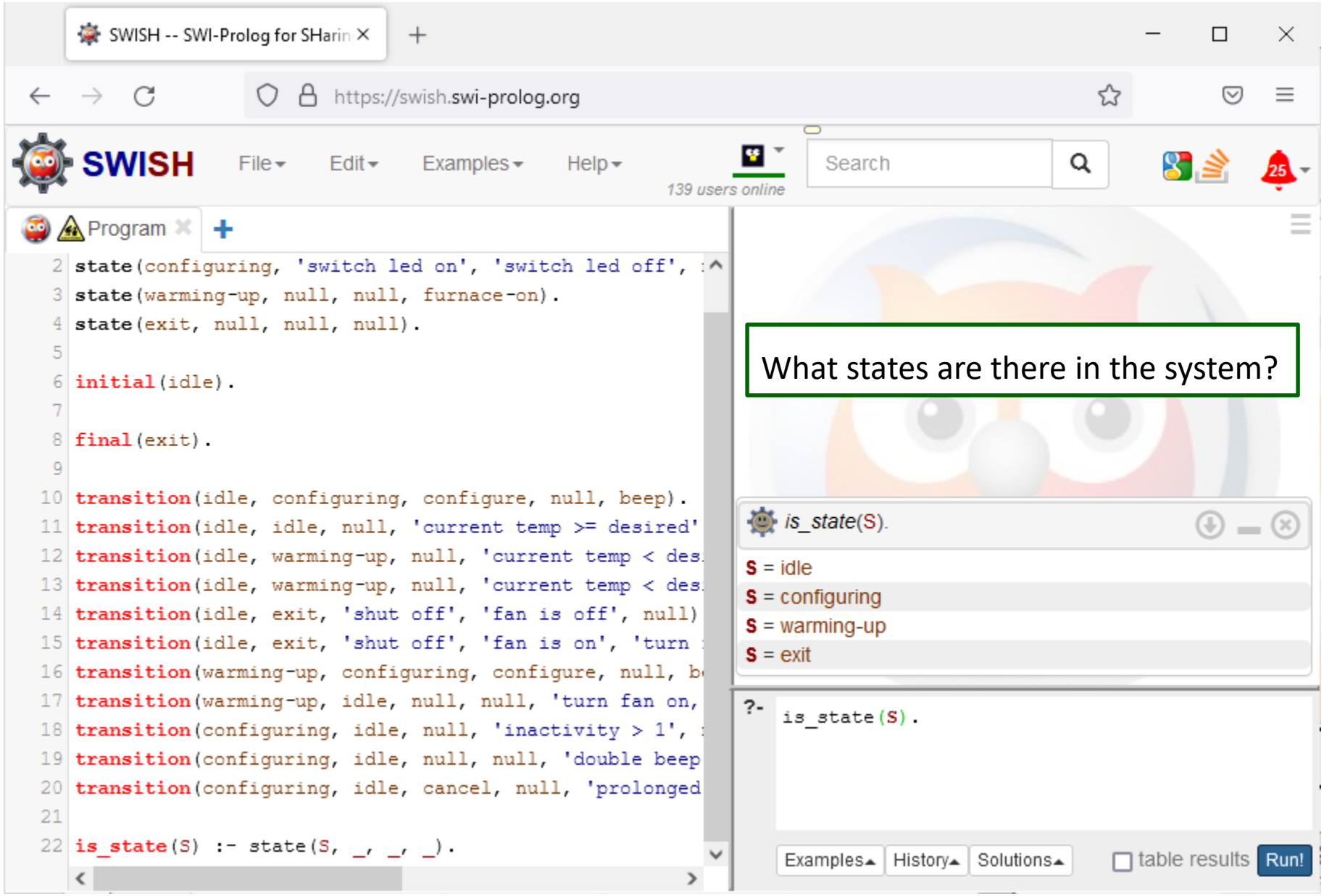

```
transition(SourceState, TargetState, Event, Guard, Action).

transition(idle, configuring, configure, null, beep).
```

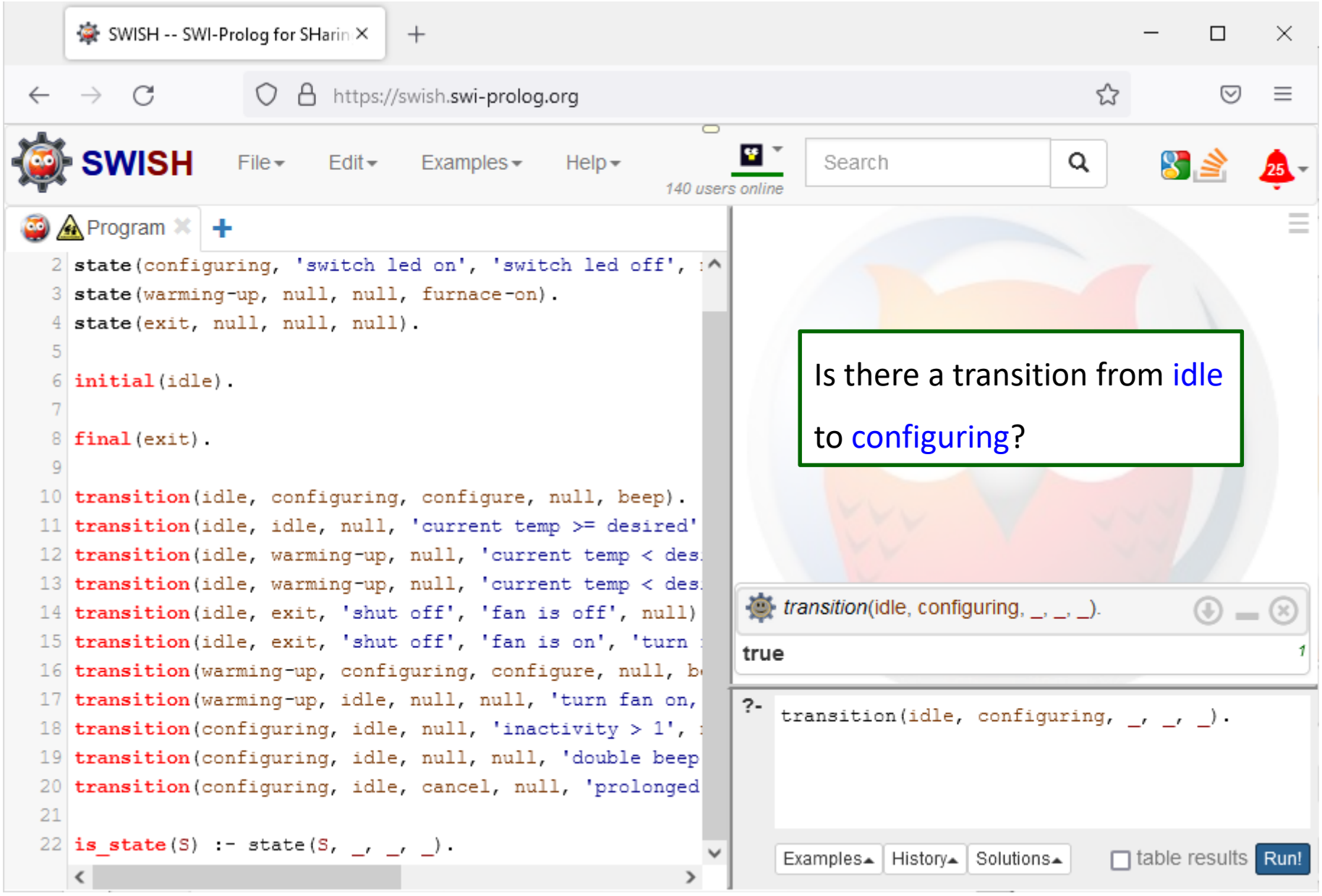

SWISH -- SWI-Prolog for SHarin
https://swish.swi-prolog.org
SWISH
File
Edit
Examples
Help
161 users online
Search
25

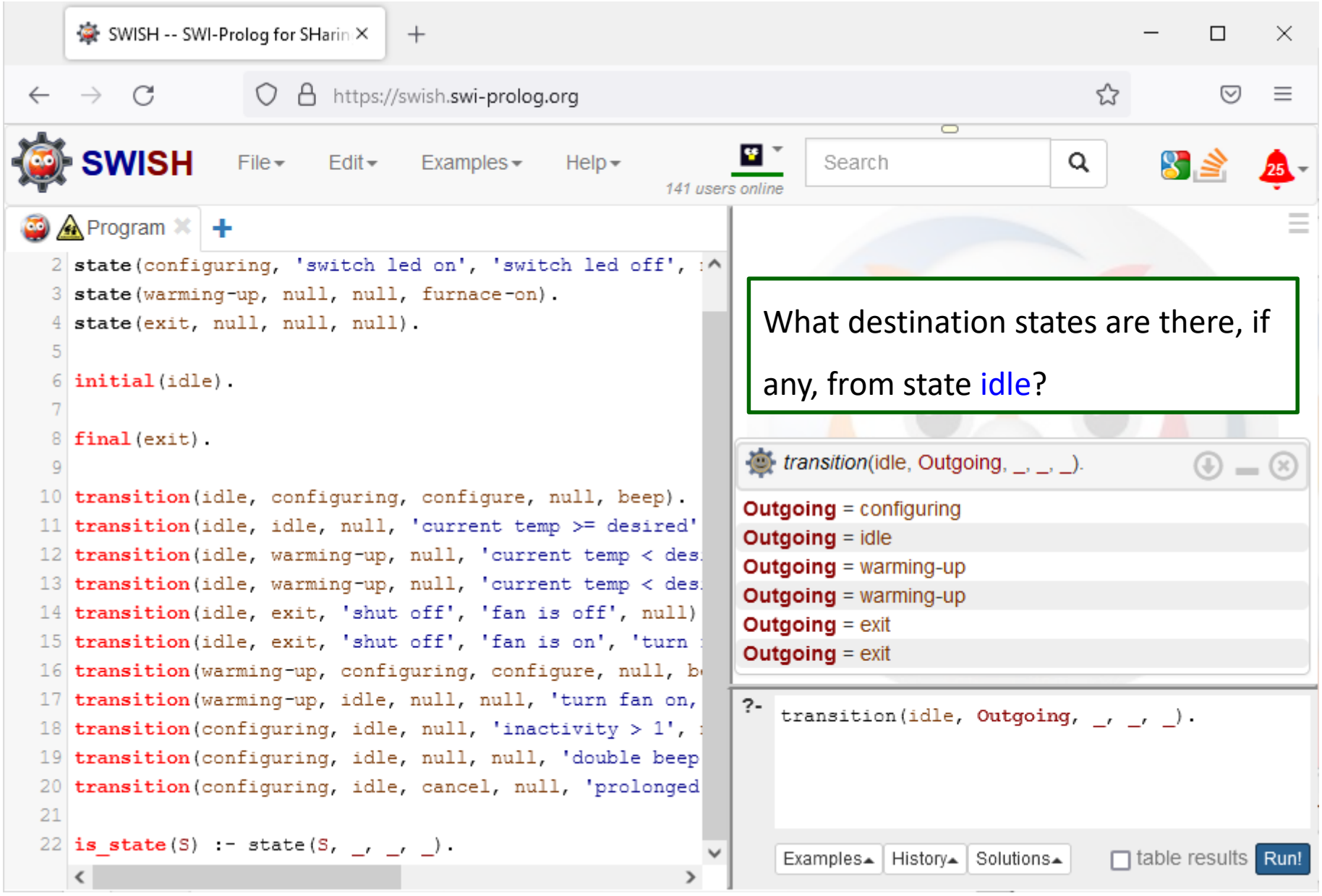

SWIS -- SWI-Prolog for SHarin
https://swish.swi-prolog.org
SWISH
File
Edit
Examples
Help
134 users online
Search
Program
1 state(idle, null, null, null).
2 state(configuring, 'switch led on', 'switch led off', nu
3 state(warming-up, null, null, furnace-on).
4 state(exit, null, null, null).
5
6 initial(idle).
7
8 final(exit).
9
10 transition(idle, configuring, configure, null, beep).
11 transition(idle, idle, null, 'current temp >= desired',
12 transition(idle, warming-up, null, 'current temp < desir
13 transition(idle, warming-up, null, 'current temp < desir
14 transition(idle, exit, 'shut off', 'fan is off', null).
15 transition(idle, exit, 'shut off', 'fan is on', 'turn fa
16 transition(warming-up, configuring, configure, null, bee
17 transition(warming-up, idle, null, null, 'turn fan on, c
18 transition(configuring, idle, null, 'inactivity > 1', nu
19 transition(configuring, idle, null, null, 'double beep')
20 transition(configuring, idle, cancel, null, 'prolonged b
Is 'idle' a state in the system?
state(idle, _, _, _).
true
1
?- state(idle, _, _, _).
Examples
History
Solutions
table results
Run!

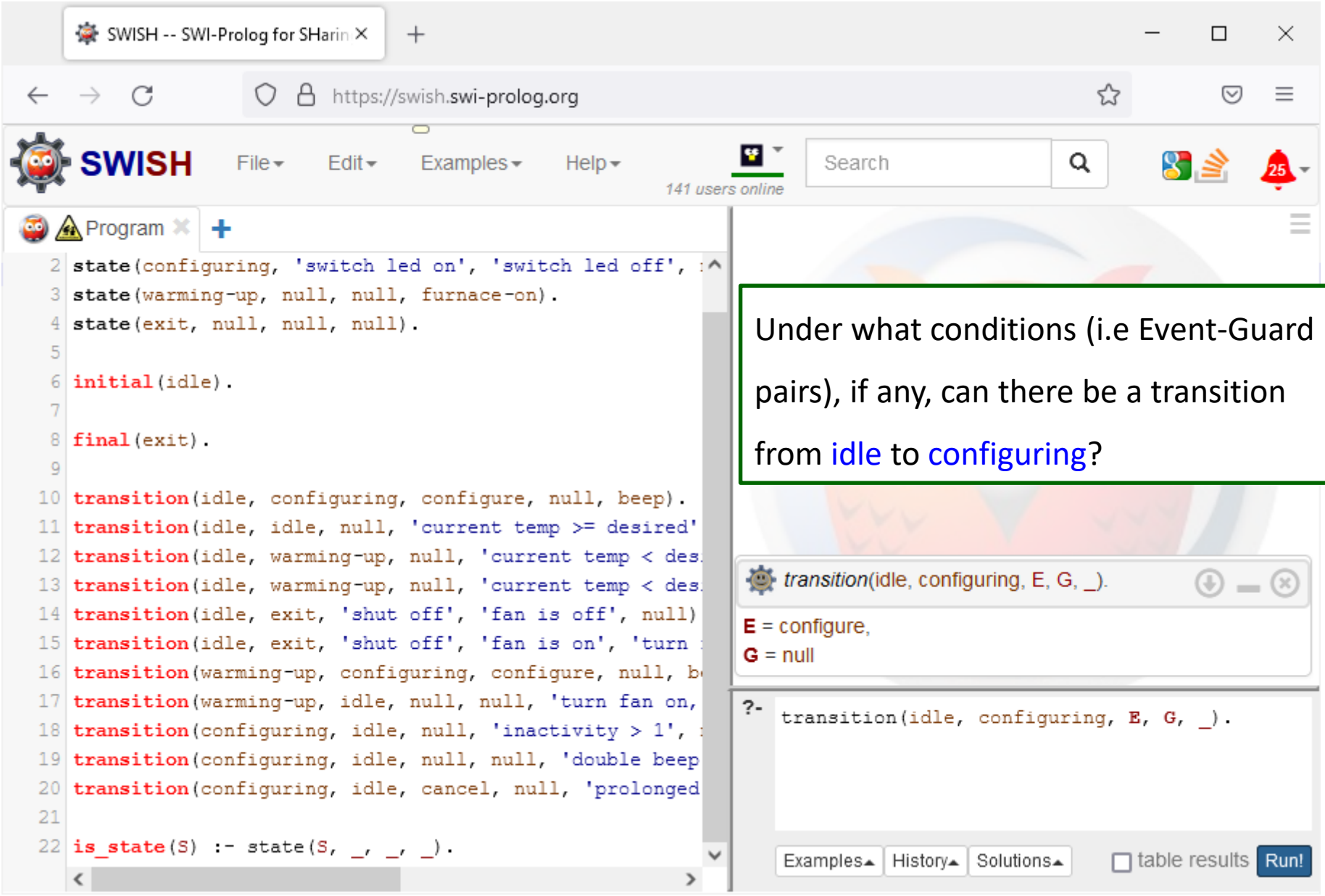

SWISH -- SWI-Prolog for SHarin
https://swish.swi-prolog.org
SWISH
File
Edit
Examples
Help
137 users online
Search
Program
state(configuring, 'switch led on', 'switch led off',
state(warming-up, null, null, furnace-on).
state(exit, null, null, null).
initial(idle).
final(exit).
transition(idle, configuring, configure, null, beep).
transition(idle, idle, null, 'current temp >= desired'
transition(idle, warming-up, null, 'current temp < des
transition(idle, warming-up, null, 'current temp < des
transition(idle, exit, 'shut off', 'fan is off', null)
transition(idle, exit, 'shut off', 'fan is on', 'turn
transition(warming-up, configuring, configure, null, b
transition(warming-up, idle, null, null, 'turn fan on,
transition(configuring, idle, null, 'inactivity > 1',
transition(configuring, idle, null, null, 'double beep
transition(configuring, idle, cancel, null, 'prolonged
is_state(S) :- state(S, _, _, _).
Adding a rule.
Is 'idle' a state in the system?
is_state(idle).
true
?- is_state(idle).
Examples
History
Solutions
table results
Run!


SWISH -- SWI-Prolog for SHarin
https://swish.swi-prolog.org
SWISH
File
Edit
Examples
Help
140 users online
Search
Program
state(configuring, 'switch led on', 'switch led off',
state(warming-up, null, null, furnace-on).
state(exit, null, null, null).
initial(idle).
final(exit).
transition(idle, configuring, configure, null, beep).
transition(idle, idle, null, 'current temp >= desired'
transition(idle, warming-up, null, 'current temp < des
transition(idle, warming-up, null, 'current temp < des
transition(idle, exit, 'shut off', 'fan is off', null)
transition(idle, exit, 'shut off', 'fan is on', 'turn
transition(warming-up, configuring, configure, null, b
transition(warming-up, idle, null, null, 'turn fan on,
transition(configuring, idle, null, 'inactivity > 1',
transition(configuring, idle, null, null, 'double beep
transition(configuring, idle, cancel, null, 'prolonged
is_state(S) :- state(S, _, _, _).
What states are there in the system?
is_state(S).
S = idle
Next
10
100
1,000
Stop
?-
is_state(S).
Examples
History
Solutions
table results
Run!


SWISH -- SWI-Prolog for SHarin
https://swish.swi-prolog.org
SWISH
File
Edit
Examples
Help
139 users online
Search
25
Program
2 state(configuring, 'switch led on', 'switch led off',
3 state(warming-up, null, null, furnace-on).
4 state(exit, null, null, null).
5
6 initial(idle).
7
8 final(exit).
9
10 transition(idle, configuring, configure, null, beep).
11 transition(idle, idle, null, 'current temp >= desired'
12 transition(idle, warming-up, null, 'current temp < des
13 transition(idle, warming-up, null, 'current temp < des
14 transition(idle, exit, 'shut off', 'fan is off', null)
15 transition(idle, exit, 'shut off', 'fan is on', 'turn
16 transition(warming-up, configuring, configure, null, b
17 transition(warming-up, idle, null, null, 'turn fan on,
18 transition(configuring, idle, null, 'inactivity > 1',
19 transition(configuring, idle, null, null, 'double beep
20 transition(configuring, idle, cancel, null, 'prolonged
21
22 is_state(S) :- state(S, _, _, _).
What states are there in the system?
is_state(S).
S = idle
S = configuring
S = warming-up
S = exit
?- is_state(S).
Examples
History
Solutions
table results
Run!


SWISH -- SWI-Prolog for SHarin
https://swish.swi-prolog.org
SWISH
File
Edit
Examples
Help
140 users online
Search
Program
2 state(configuring, 'switch led on', 'switch led off', :
3 state(warming-up, null, null, furnace-on).
4 state(exit, null, null, null).
5
6 initial(idle).
7
8 final(exit).
9
10 transition(idle, configuring, configure, null, beep).
11 transition(idle, idle, null, 'current temp >= desired'
12 transition(idle, warming-up, null, 'current temp < des
13 transition(idle, warming-up, null, 'current temp < des
14 transition(idle, exit, 'shut off', 'fan is off', null)
15 transition(idle, exit, 'shut off', 'fan is on', 'turn
16 transition(warming-up, configuring, configure, null, b
17 transition(warming-up, idle, null, null, 'turn fan on,
18 transition(configuring, idle, null, 'inactivity > 1',
19 transition(configuring, idle, null, null, 'double beep
20 transition(configuring, idle, cancel, null, 'prolonged
21
22 is_state(S) :- state(S, _, _, _).
Is there a transition from idle to configuring?
transition(idle, configuring, _, _, _).
true
1
?- transition(idle, configuring, _, _, _).
Examples
History
Solutions
table results
Run!


SWISH -- SWI-Prolog for SHarin
https://swish.swi-prolog.org
SWISH
File
Edit
Examples
Help
141 users online
Search
Program
2 state(configuring, 'switch led on', 'switch led off', :
3 state(warming-up, null, null, furnace-on).
4 state(exit, null, null, null).
5
6 initial(idle).
7
8 final(exit).
9
10 transition(idle, configuring, configure, null, beep).
11 transition(idle, idle, null, 'current temp >= desired'
12 transition(idle, warming-up, null, 'current temp < des
13 transition(idle, warming-up, null, 'current temp < des
14 transition(idle, exit, 'shut off', 'fan is off', null)
15 transition(idle, exit, 'shut off', 'fan is on', 'turn
16 transition(warming-up, configuring, configure, null, b
17 transition(warming-up, idle, null, null, 'turn fan on,
18 transition(configuring, idle, null, 'inactivity > 1',
19 transition(configuring, idle, null, null, 'double beep
20 transition(configuring, idle, cancel, null, 'prolonged
21
22 is_state(S) :- state(S, _, _, _).
What destination states are there, if any, from state idle?
transition(idle, Outgoing, _, _, _).
Outgoing = configuring
Outgoing = idle
Outgoing = warming-up
Outgoing = warming-up
Outgoing = exit
Outgoing = exit
?- transition(idle, Outgoing, _, _, _).
Examples
History
Solutions
table results
Run!


SWISH -- SWI-Prolog for SHarin
https://swish.swi-prolog.org
SWISH
File
Edit
Examples
Help
141 users online
Search
Program
state(configuring, 'switch led on', 'switch led off',
state(warming-up, null, null, furnace-on).
state(exit, null, null, null).
initial(idle).
final(exit).
transition(idle, configuring, configure, null, beep).
transition(idle, idle, null, 'current temp >= desired'
transition(idle, warming-up, null, 'current temp < des
transition(idle, warming-up, null, 'current temp < des
transition(idle, exit, 'shut off', 'fan is off', null)
transition(idle, exit, 'shut off', 'fan is on', 'turn
transition(warming-up, configuring, configure, null, b
transition(warming-up, idle, null, null, 'turn fan on,
transition(configuring, idle, null, 'inactivity > 1',
transition(configuring, idle, null, null, 'double beep
transition(configuring, idle, cancel, null, 'prolonged
is_state(S) :- state(S, _, _, _).
Under what conditions (i.e Event-Guard pairs), if any, can there be a transition from idle to configuring?
transition(idle, configuring, E, G, _).
E = configure,
G = null
?- transition(idle, configuring, E, G, _).
Examples
History
Solutions
table results
Run!

Built-in utility functions in Prolog

- The built-in function `findall(X, P, L)` returns a list `L` with all values for `X` that satisfy predicate `P`.

- To eliminate redundancies in a list, we can use the built-in function `list_to_set(List, Set)` that converts the list (with possibly repeated elements) into a set.

```
%%  events/1: Succeeds by obtaining a collection of
%%            all events that occur in the system.
events(EventSet) :-

    findall(Event,

           (transition(_, _, Event, _, _), (Event \= 'null')),

           EventList),

    list_to_set(EventList, EventSet).
```

```
%%  events/1: Succeeds by obtaining a collection of
%%            all events that occur in the system.
events(EventSet) :-

    findall(Event,

            (transition(_, _, Event, _, _), (Event \= 'null')),

            EventList),

    list_to_set(EventList, EventSet).
```

1. Given this predicate…

```
%%  events/1: Succeeds by obtaining a collection of
%%            all events that occur in the system.
events(EventSet) :-

    findall(Event,

            (transition(_, _, Event, _, _), (Event \= 'null')),

            EventList),

    list_to_set(EventList, EventSet).
```

2. Obtain all instances of Event that satisfy the predicate, and…

1. Given this predicate…

```
%%  events/1: Succeeds by obtaining a collection of
%%            all events that occur in the system.
events(EventSet) :-

    findall(Event,

            (transition(_, _, Event, _, _), (Event \= 'null')),

            EventList),

    list_to_set(EventList, EventSet).
```

2. Obtain all instances of Event that satisfy the predicate, and…

1. Given this predicate…

3. Place all those matching instances in a list.

```
%%  events/1: Succeeds by obtaining a collection of
%%            all events that occur in the system.
events(EventSet) :-

    findall(Event,

            (transition(_, _, Event, _, _), (Event \= 'null')),

            EventList),

    list_to_set(EventList, EventSet).
```

2. Obtain all instances of Event that satisfy the predicate, and…

1. Given this predicate…

3. Place all those matching instances in a list.

4. Transform the list into a set, and, …

```
%%  events/1: Succeeds by obtaining a collection of
%%            all events that occur in the system.
events(EventSet) :-

    findall(Event,

            (transition(_, _, Event, _, _), (Event \= 'null')),

            EventList),

    list_to_set(EventList, EventSet).
```

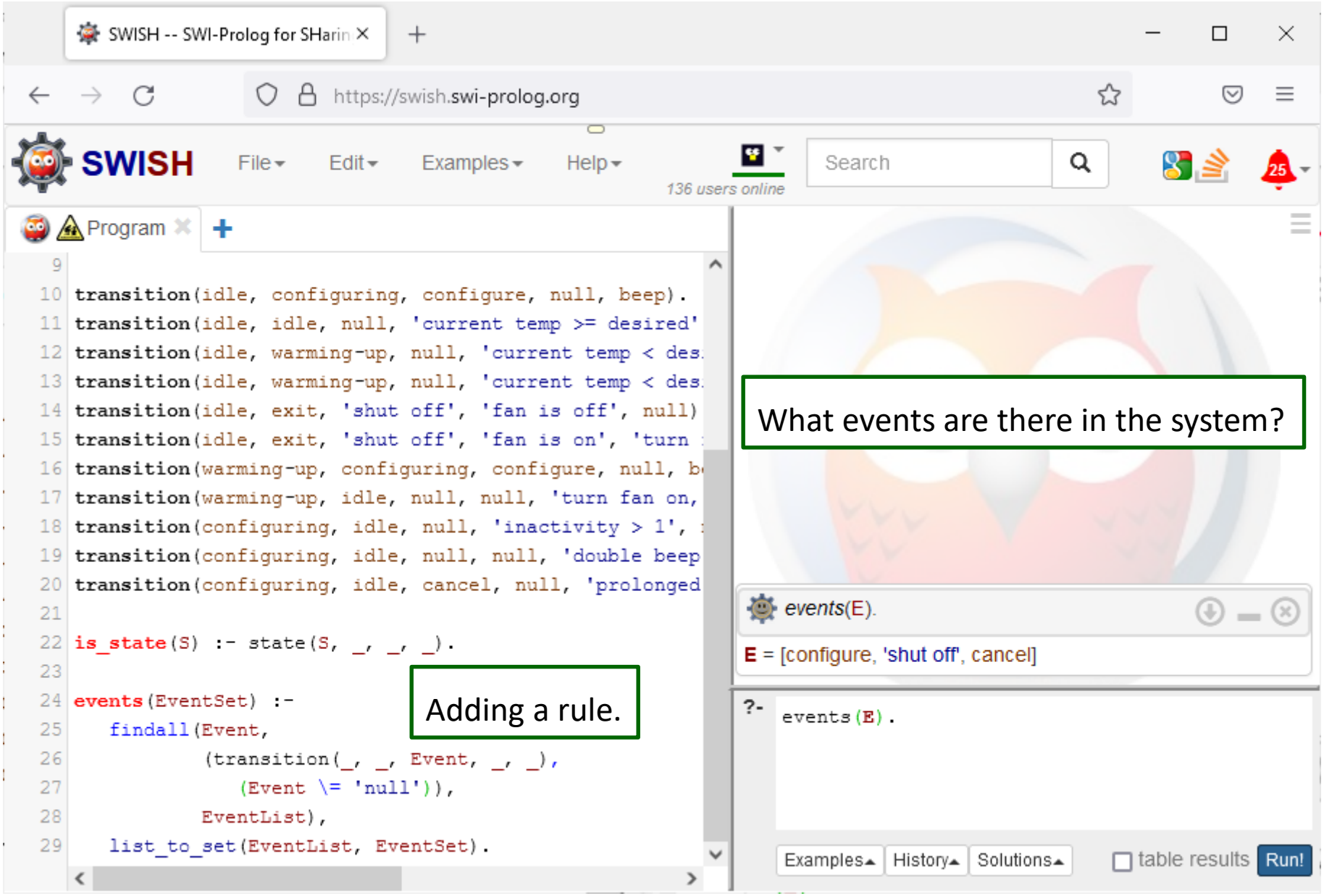
5. Return the set.

2. Obtain all instances of Event that satisfy the predicate, and…

1. Given this predicate…

3. Place all those matching instances in a list.

4. Transform the list into a set, and, …


SWISH -- SWI-Prolog for SHarin
https://swish.swi-prolog.org
SWISH
File
Edit
Examples
Help
136 users online
Search
Program
transition(idle, configuring, configure, null, beep).
transition(idle, idle, null, 'current temp >= desired'
transition(idle, warming-up, null, 'current temp < des
transition(idle, warming-up, null, 'current temp < des
transition(idle, exit, 'shut off', 'fan is off', null)
transition(idle, exit, 'shut off', 'fan is on', 'turn
transition(warming-up, configuring, configure, null, b
transition(warming-up, idle, null, null, 'turn fan on,
transition(configuring, idle, null, 'inactivity > 1',
transition(configuring, idle, null, null, 'double beep
transition(configuring, idle, cancel, null, 'prolonged
is_state(S) :- state(S, _, _, _).
events(EventSet) :-
findall(Event,
(transition(_, _, Event, _, _),
(Event \= 'null')),
EventList),
list_to_set(EventList, EventSet).
Adding a rule.
What events are there in the system?
events(E).
E = [configure, 'shut off', cancel]
?-
events(E).
Examples
History
Solutions
table results
Run!

```
%%  get_transitions/1 :  The rule succeeds
%%     by returning a set of non-self transition pairs of the
%%     form [SourceState, DestinationState].
get_transitions(T) :-

     findall([S, D],

                (transition(S, D, _, _, _),

                S \== D),

                L),

     list_to_set(L, T).
```

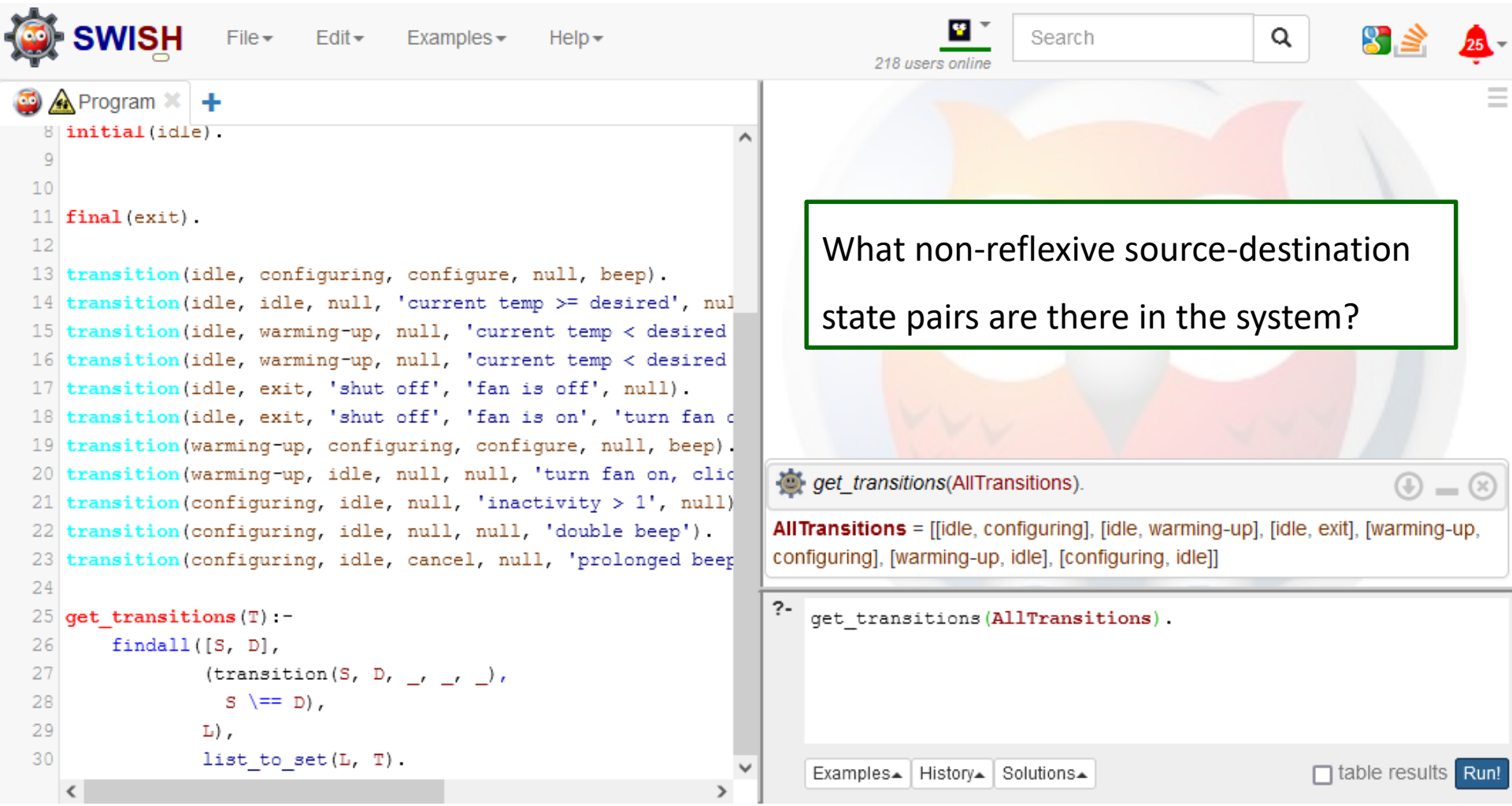
5. Return the set.

2. Obtain all instances of Source Destination pairs that satisfy the predicate, and…

1. Given this predicate…

3. Place all those matching instances in a list.

4. Transform the list into a set, and, …


SWISH
File
Edit
Examples
Help
218 users online
Search
25
Program
initial(idle).
final(exit).
transition(idle, configuring, configure, null, beep).
transition(idle, idle, null, 'current temp >= desired', nul
transition(idle, warming-up, null, 'current temp < desired
transition(idle, warming-up, null, 'current temp < desired
transition(idle, exit, 'shut off', 'fan is off', null).
transition(idle, exit, 'shut off', 'fan is on', 'turn fan c
transition(warming-up, configuring, configure, null, beep).
transition(warming-up, idle, null, null, 'turn fan on, clic
transition(configuring, idle, null, 'inactivity > 1', null)
transition(configuring, idle, null, null, 'double beep').
transition(configuring, idle, cancel, null, 'prolonged beep
get_transitions(T):-
findall([S, D],
(transition(S, D, _, _, _),
S \== D),
L),
list_to_set(L, T).
What non-reflexive source-destination state pairs are there in the system?
get_transitions(AllTransitions).
AllTransitions = [[idle, configuring], [idle, warming-up], [idle, exit], [warming-up, configuring], [warming-up, idle], [configuring, idle]]
?-
get_transitions(AllTransitions).
Examples
History
Solutions
table results
Run!