In [1]:
```julia
using Pkg
Pkg.activate(joinpath("..", "environments", "modal-association-rules"))
Pkg.instantiate()
```

```
  Activating project at `~/logic-and-machine-learning/environments/modal-a
ssociation-rules`
```

In [2]:
```julia
using Plots
using Random
using Statistics

Random.seed!(1605)
```

Out[2]:
```
TaskLocalRNG()
```

In [3]:
```julia
# this will be useful later, for loading the Natops dataset
function parse_natops(arffstring::String)
    df = DataFrame()
    classes = String[]

    lines = split(arffstring, "\n")
    for i in 1:length(lines)
        line = lines[i]

        # split the current line;
        # if it is not a data line, starting with DATA_MARK, continue;
        # continue even in the case where checking the first character th
        # out an error.
        sline = nothing
        try
            sline = split(line, " ")
            if sline[1][1] != '\''
                continue
            end
        catch
            continue
        end

        # skip the initial hypen an read the data
        sline[1] = sline[1][2:end]
        data_and_class = split(sline[1], "\'")
        string_data = split(data_and_class[1], "\\n")
        class = data_and_class[2][2:end]

        if isempty(names(df))
            for i in 1:length(string_data)
                insertcols!(df, Symbol("V$(i)") => Array{Float64, 1}[])
            end
        end

        float_data = Dict{Int,Vector{Float64}}()

        for i in 1:length(string_data)
            float_data[i] = map(x->parse(Float64,x), split(string_data[i]
        end

        push!(df, [float_data[i] for i in 1:length(string_data)])
        push!(classes, class)
```

```
    end

    p = sortperm(eachrow(df), by=x->classes[rownumber(x)])

    return df[p, :], classes[p]
end
```

Out[3]: `parse_natops (generic function with 1 method)`

Modal Association Rules

In this notebook, we follow a different paradigm with respect to the supervised one we saw in day04-learning-with-non-classical-logical.

The hypothesis here, is that a logical formula is *interesting*, if it happens to be frequently satisfied across all the instances of a dataset $\mathcal{I}$.

Given an alphabet $\mathcal{P}$ of propositional literals, the formula we are dealing with are literal conjunctions called *itemsets*.

An itemset that is also frequent is called *frequent itemset*.

More formally, given a dataset $\mathcal{I}$, a propositional alphabet $\mathcal{P}$ and a minimum threshold s, a frequent pattern $\mathsf{P} \subseteq \mathcal{P}$ is such that:

$$\mathrm{support}(\mathcal{I}, \mathsf{P}) = \frac{|\{I \in \mathcal{I} \mid I \models \mathsf{P}\}|}{|\mathcal{I}|} \geq s$$

The ratio above is called *support*.

In [4]:
```
using ModalAssociationRules
```

In [5]:
```
# these are just three toy atoms
p = Atom(ScalarCondition(VariableMax(4), >=, 2)) |> Item
q = Atom(ScalarCondition(VariableMin(5), <=, 1.5)) |> Item
r = Atom(ScalarCondition(VariableMax(6), >=, 0.0)) |> Item
```

Out[5]: `max[V6] ≥ 0.0`

In [6]:
```
# an Itemset encodes a conjunction of SoleLogics.Formula,
# but has two advantages:

# 1) performance considerations
# https://towardsdev.com/set-vs-vector-lookup-in-julia-a-closer-look-9d10
# 2) type piracy prevention!

pq = Itemset([p, q])
pr = Itemset([p, r])
qr = Itemset([q, r])
pqr = Itemset([p, q, r])
```

Out[6]:
```
3-element Vector{Item}:
 max[V4] ≥ 2
 min[V5] ≤ 1.5
 max[V6] ≥ 0.0
```

In [7]:
```julia
# an Itemset can wrap any SoleLogics.Formula type;
formula(pq)
```

Out[7]:
```
LeftmostConjunctiveForm with 2 Atom grandchildren:
        max[V4] ≥ 2
        min[V5] ≤ 1.5
```

Exercise:

Define your own `mysupport` function.

Its argument must be of type `SoleLogics.Formula`, `SoleData.AbstractLogiset` and `SoleLogics.AbstractWorld`.

We only want to consider the instances that were originally associated with the `I have command` class.

We want to treat the Kripke model as a degenerate propositional logiset.

Then compute the support of the following itemsets: `p`, `q`, `r`, `p ∧ q`, `p ∧ r`, `r ∧ q`, `p ∧ q ∧ r`.

The support must be rounded to the second decimal digit.

Solution (Base64):
ZnVuY3Rpb24gbXlzdXBwb3J0KHBoaTo6RiwgWGs6OkwsIHdvcmxkOjpXKSB3aGVyZSB7C

In [8]:
```julia
# Write your definition here
```

In [9]:
```julia
try
    for phi in [p, q, r, pq, pr, qr, pqr]
        println(
            mysupport(
                formula(phi),
                SoleData.slicedataset(Xk, 1:30),
                Interval(1, X_ndatapoints)
            )
        )
    end
catch e
    if e isa UndefVarError
        println("You need to implement mysupport.")
    end
end
```

```
You need to implement mysupport.
```

Let us consider an alphabet of propositional literals $\mathcal{P}$, and let us suppose that $\mathrm{P} \subseteq \mathcal{P}$ is a frequent pattern we found.

We can partition P in two smaller frequent patterns, Q, R, such that $\mathrm{Q} \cap \mathrm{R} = \emptyset$.

We denote with $\mathsf{Q} \Rightarrow \mathsf{R}$ the fact that an *interesting* statistical relation occurs between the antecedent and the consequent: if this is the case, then we have an *association rule*.

Similarly to the case of frequent patterns, the interestingness must be established with specific measures, which are called *meaningfulness measures* in the jargon.

In [10]:
```julia
# beware of the difference between an Item (such as p) and an Itemset;
# we need to cast p to Itemset, even if it is a trivial 1-length Itemset.
println(typeof(p))
ARule(Itemset(p), qr)
```

```
Item{Atom{ScalarCondition{Int64, VariableMax{Int64}, ScalarMetaCondition{VariableMax{Int64}, typeof(>=)}}}}
```

Out[10]:
```
max[V4] ≥ 2 => (min[V5] ≤ 1.5) ∧ (max[V6] ≥ 0.0)
```

In [11]:
```julia
try
    ARule(pq, qr)
catch e
    if e isa ArgumentError
        println("Beware: pq ∩ qr is not empty.")
    end
end
```

```
Beware: pq ∩ qr is not empty.
```

In [12]:
```julia
rule = ARule(Itemset(p), qr)
```

Out[12]:
```
max[V4] ≥ 2 => (min[V5] ≤ 1.5) ∧ (max[V6] ≥ 0.0)
```

In [13]:
```julia
ModalAssociationRules.antecedent(rule)
```

Out[13]:
```
1-element Vector{Item{Atom{ScalarCondition{Int64, VariableMax{Int64}, ScalarMetaCondition{VariableMax{Int64}, typeof(>=)}}}}}:
 max[V4] ≥ 2
```

In [14]:
```julia
ModalAssociationRules.consequent(rule)
```

Out[14]:
```
2-element Vector{Item}:
 min[V5] ≤ 1.5
 max[V6] ≥ 0.0
```

In [15]:
```julia
# get the generator Itemset back
Itemset(rule)
```

Out[15]:
```
3-element Vector{Item}:
 max[V4] ≥ 2
 min[V5] ≤ 1.5
 max[V6] ≥ 0.0
```

Quiz

Try to explain the ratio below, which is commonly called *confidence*.

$$\text{confidence}(\mathcal{I}, \mathsf{P} \Rightarrow \mathsf{Q}) = \frac{\text{support}(\mathcal{I}, \mathsf{P} \cap \mathsf{Q})}{\text{support}(\mathcal{I}, \mathsf{P})}$$

Exercise

Implement your own `myconfidence` function.

Solution (Base 64):
ZnVuY3Rpb24gbXljb25maWRlbmNlKHJ1bGU6OkFSdWxlLCBYazo6TCwgd29ybGQ6OlcpIH

In [16]:
```
# Insert your solution here
```

In [17]:
```
try
    for phi in [p, q, r, pq, pr, qr, pqr]
        println(
            myconfidence(
                rule,
                SoleData.slicedataset(Xk, 1:30),
                Interval(1, X_ndatapoints)
            )
        )
    end
catch e
    if e isa UndefVarError
        println("You need to implement myconfidence.")
    end
end
```

```
You need to implement myconfidence.
```

Enhancing Modal Association Rules with Modalities

When dealing with Kripke models, a natural dichotomy pops up!

Let us consider an alphabet of modal literals $\Lambda_{\mathcal{P}}$, obtained by enriching a standard, propositional alphabet $\mathcal{P}$ with modal operators.

Let us also consider a modal dataset $\mathcal{I}$ and an instance $I = (W, R, v) \in \mathcal{I}$ in it, as well as a pattern P.

We can assess the interestingness of P within an instance by computing its *local support*, and comparing it with respect to a *minimum local support threshold* s_l.

$$\text{lsupport}(I, \mathsf{P}) = \frac{|\{w \in W \mid I, w \models \mathsf{P}\}|}{|\mathcal{W}|}$$

The other part of the dichotomy, that is, the notion of *global* support, is left as an exercise (see the Quiz below).

In [18]:
```
lsupport
```

Out[18]:
```
lsupport (generic function with 1 method)
```

In [19]:
```
gsupport
```

Out[19]:
```
gsupport (generic function with 1 method)
```

In [20]:
```
lconfidence
```

Out[20]: lconfidence (generic function with 1 method)

In [21]:
```
gconfidence
```

Out[21]: gconfidence (generic function with 1 method)

Quiz

How would you aggregate many local support computations, to compute a *global* support?

Mining Association Rules from Time Series Items

We want to probe our instances with considerations on the shape of the signal in a certain interval, for a given feature.

We also want to increase the expressiveness of the result association rules with the help of HS logic.

In [22]:
```
X, y = read(
    joinpath(@__DIR__, "..", "datasets", "natops.arff"), String) |> parse
```

Out[22]:

```
(360×24 DataFrame
 Row │ V1                                  V2
V …
     │ Array…                              Array…
A …
─────┼─────────────────────────────────────────────────────────────────
────────
   1 │ [-0.519771, -0.52758, -0.531415,…  [-2.14011, -2.18043, -2.18425,
-…  [ …
   2 │ [-0.489753, -0.48607, -0.484529,…  [-1.55293, -1.54966, -1.55206,
-…  [
   3 │ [-0.521346, -0.518394, -0.522321…  [-1.72326, -1.72407, -1.72326,
-…  [
   4 │ [-0.57022, -0.562064, -0.565967,…  [-1.91196, -1.90369, -1.90527,
-…  [
   5 │ [-0.624417, -0.626031, -0.625388…  [-1.84287, -1.84026, -1.84688,
-…  [ …
   6 │ [-0.502501, -0.502525, -0.499415…  [-2.17556, -2.15613, -2.18516,
-…  [
   7 │ [-0.488461, -0.489463, -0.487539…  [-2.17242, -2.18203, -2.18057,
-…  [
   8 │ [-0.468105, -0.410602, -0.473909…  [-1.86535, -1.89011, -1.87105,
-…  [
   9 │ [-0.568195, -0.572936, -0.571337…  [-1.79059, -1.78162, -1.78303,
-…  [ …
  10 │ [-0.517579, -0.515374, -0.517325…  [-1.73887, -1.74072, -1.7397,
-1…  [
  11 │ [-0.631494, -0.629032, -0.630474…  [-1.98071, -1.98581, -1.98407,
-…  [
  ⋮  │                 ⋮                                  ⋮
⋱
 351 │ [-0.937442, -1.02995, -0.985338,…  [-2.10495, -1.96835, -1.95082,
-…  [
 352 │ [-0.595625, -0.593605, -0.588495…  [-1.93897, -1.93641, -1.93761,
-…  [ …
 353 │ [-0.466582, -0.469372, -0.43454,…  [-1.71874, -1.72861, -1.58691,
-…  [
 354 │ [-0.500404, -0.502824, -0.504771…  [-1.89209, -1.88986, -1.89188,
-…  [
 355 │ [-0.686893, -0.690966, -0.710514…  [-2.04375, -2.05011, -2.07035,
-…  [
 356 │ [-0.525938, -0.516073, -0.5177, …  [-1.69259, -1.70519, -1.70824,
-…  [ …
 357 │ [-0.440887, -0.452221, -0.447185…  [-1.90444, -1.91917, -1.91652,
-…  [
 358 │ [-0.647672, -0.653511, -0.642305…  [-1.6173, -1.61051, -1.60491,
-1…  [
 359 │ [-0.476117, -0.4705, -0.474443, …  [-1.70846, -1.7099, -1.70912,
-1…  [
 360 │ [-0.553245, -0.551704, -0.548044…  [-1.69493, -1.69349, -1.69951,
-…  [ …
                                                  22 columns and 339 rows
omitted, ["1.0", "1.0", "1.0", "1.0", "1.0", "1.0", "1.0", "1.0", "1.0",
"1.0"  …  "6.0", "6.0", "6.0", "6.0", "6.0", "6.0", "6.0", "6.0", "6.0",
"6.0"])
```

In [23]:

```julia
function _normalize(x::Vector{R}) where {R <: Real}
    eps = 1e-10
    return (x .- mean(x)) ./ (std(x) + eps)
end
```

```julia
function zeuclidean(x::Vector{R}, y::Vector{R}) where {R}
    # normalize x and y
    meanx = mean(x)
    meany = mean(y)

    # avoid division by zero
    eps = 1e-10

    x_z = _normalize(x)
    y_z = _normalize(y)

    # z-normalized euclidean distance formula
    return sqrt(sum((x_z .- y_z).^2))
end
```

Out[23]: zeuclidean (generic function with 1 method)

In [24]:
```julia
# consider only right hand and right elbow
varids = vcat(collect(4:6), collect(10:12));
```

In [25]:
```julia
mar_res_path = joinpath(@__DIR__, "..", "other-resources", "natops-for-ma
```

Out[25]: "/home/alberto-paparella/logic-and-machine-learning/notebooks/../other-resources/natops-for-mar"

In [26]:
```julia
using Serialization

function load_motifs(filepath, save_filename_prefix)
    ids = [
        id for id in deserialize(
            joinpath(filepath, "$(save_filename_prefix)-ids")
        )
    ];
    motifs = [
        m for m in deserialize(
            joinpath(filepath, "$(save_filename_prefix)-motifs")
        )
    ];
    featurenames = [
        f for f in deserialize(
            joinpath(filepath, "$(save_filename_prefix)-featurenames")
        )
    ];
    return ids, motifs, featurenames
end


ids, motifs, featurenames = load_motifs(mar_res_path, "NATOPS-IHCC");
```

In this example, we only consider intervals of length 10 and 20.

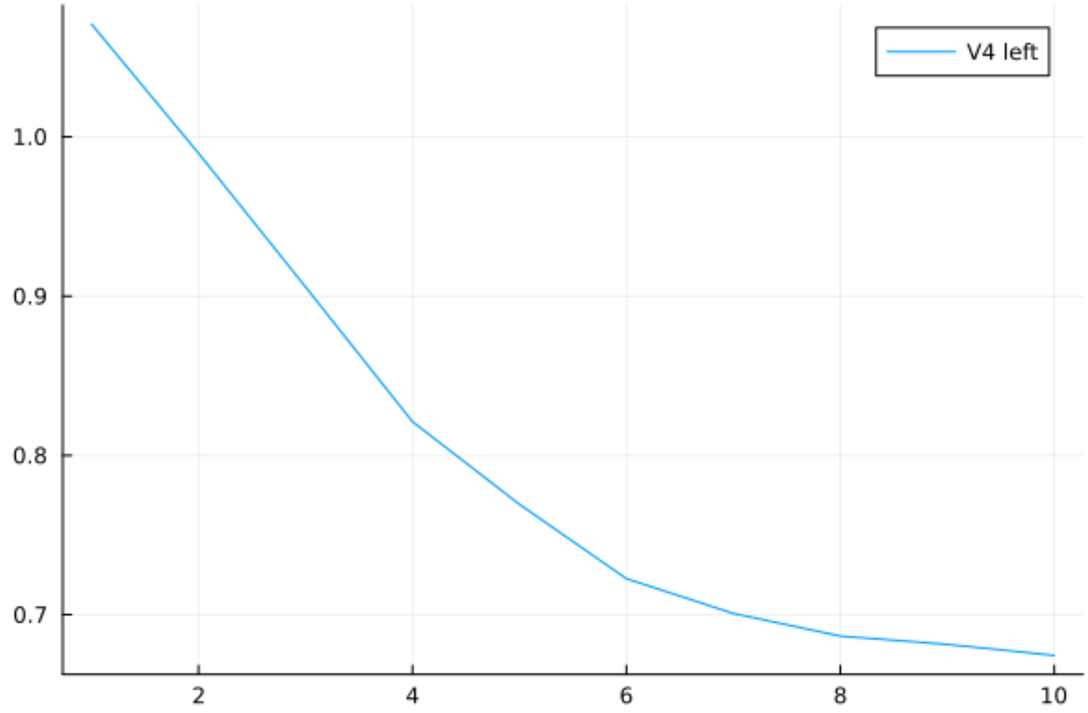
In particular, given a world encoding an interval of such length, we compute the (normalized) euclidean distance between it and a pool of particular time series, called "motifs".

If the distance is low enough, then it means that the gesture encoded by the motif is happening.

Try to browse the motifs we are playing with, by tweaking the plot below.

In [27]:
```
i = 1
plot(motifs[i], label = "V$(ids[i]) $(featurenames[i])")
```

Out[27]:



In [28]:
```
_variables = [
    SoleData.VariableDistance(id, m, distance=zeuclidean, featurename=nam
    for (id, m, name) in zip(ids, motifs, featurenames)
]

syntaxstring.(_variables)[1:3]
```

Out[28]:
```
3-element Vector{String}:
 "left[V4]"
 "right[V4]"
 "inv_left[V4]"
```

In [29]:
```
# we only consider the instances related to the "I have command" class;
# we are not cheating: we just want to describe the instances
IHCC = reduce(vcat, [X[1:30, :], X[(180+1):(180+30), :]]);
IHCCk = scalarlogiset(IHCC, _variables)
```

Out[29]:
```
SupportedLogiset with 1 support (50.02 MBs)
├ worldtype:                   Interval{Int64}
├ featvaltype:                 Float64
├ featuretype:                 VariableDistance{Int64, Vector{Float64}}
├ frametype:                   FullDimensionalFrame{1, Interval{Int64}}
├ # instances:                 60
├ usesfullmemo:                true
├[BASE] UniformFullDimensionalLogiset of channel size (51,) (50.02 MBs)
│ ├ size × eltype:              (51, 51, 60, 42) × Float64
│ └ features:                   42 -> VariableDistance{Int64, Vector{Flo
at64}}[left[V4], right[V4], inv_left[V4], pause_right[V4], ..., pause_fr
ont[V12], front_pause[V12], longfront[V12], longbehind[V12], pause_sligh
tlybehind_front[V12]]
└[SUPPORT 1] FullMemoset (0 memoized values, 5.25 KBs))
```

In [30]:
```
propositionalatoms = [
    Atom(ScalarCondition(v, <=, 1.0))
    for v in _variables
]

syntaxstring.(propositionalatoms)[1:3]
```

Out[30]:
```
3-element Vector{String}:
 "left[V4] ≤ 1.0"
 "right[V4] ≤ 1.0"
 "inv_left[V4] ≤ 1.0"
```

In [31]:
```
atoms = Vector{Item}(
    reduce(vcat, [
        propositionalatoms,
        diamond(IA_A).(propositionalatoms),
        diamond(IA_B).(propositionalatoms),
        diamond(IA_E).(propositionalatoms),
        diamond(IA_D).(propositionalatoms),
        diamond(IA_O).(propositionalatoms),
    ])
)

syntaxstring.(atoms)[1:3]
```

Out[31]:
```
3-element Vector{String}:
 "left[V4] ≤ 1.0"
 "right[V4] ≤ 1.0"
 "inv_left[V4] ≤ 1.0"
```

In [32]:
```
_items = Vector{Item}(atoms);
```

In [33]:
```
miner = Miner(
    # the data from which we want to find all the frequent itemsets
    IHCCk,

    # the strategy we want to leverage for exploring the frequent itemset
    apriori,

    # the initial alphabet of facts
    _items,

    # the interestingness measures for the frequent itemsets
    [(gsupport, 0.1, 0.1)],

    # the meaningfulness measures for the association rules
    [(gconfidence, 0.5, 0.5)];

    worldfilter=SoleLogics.FunctionalWorldFilter(
        x -> (length(x) == 10) || (length(x) == 20), Interval{Int}
    ),

    itemset_policies=Function[
        isanchored_itemset(ignoreuntillength=1),
        isdimensionally_coherent_itemset()
    ],

    arule_policies=Function[
        islimited_length_arule(consequent_maxlength=3),
        isanchored_arule()
```

```
    ]
)
```

```
Out[33]: SupportedLogiset with 1 support (50.02 MBs)
├ worldtype:                   Interval{Int64}
├ featvaltype:                 Float64
├ featuretype:                 VariableDistance{Int64, Vector{Float64}}
├ frametype:                   FullDimensionalFrame{1, Interval{Int64}}
├ # instances:                 60
├ usesfullmemo:                true
├[BASE] UniformFullDimensionalLogiset of channel size (51,) (50.02 MBs)
│ ├ size × eltype:              (51, 51, 60, 42) × Float64
│ └ features:                   42 -> VariableDistance{Int64, Vector{Flo
at64}}[left[V4], right[V4], inv_left[V4], pause_right[V4], ..., pause_fr
ont[V12], front_pause[V12], longfront[V12], longbehind[V12], pause_sligh
tlybehind_front[V12]]
└[SUPPORT 1] FullMemoset (0 memoized values, 5.25 KBs))

Alphabet: [left[V4] ≤ 1.0, right[V4] ≤ 1.0, inv_left[V4] ≤ 1.0, pause_ri
ght[V4] ≤ 1.0, inv_left_inv[V4] ≤ 1.0, pause_right_inv[V4] ≤ 1.0, left_i
nv_right[V4] ≤ 1.0, down[V5] ≤ 1.0, up[V5] ≤ 1.0, down_pause[V5] ≤ 1.0,
pause_up[V5] ≤ 1.0, longdown_pause[V5] ≤ 1.0, longup[V5] ≤ 1.0, longpaus
e_up[V5] ≤ 1.0, front[V6] ≤ 1.0, behind[V6] ≤ 1.0, behind_pause[V6] ≤ 1.
0, front_pause[V6] ≤ 1.0, longbehind[V6] ≤ 1.0, longfront[V6] ≤ 1.0, fro
nt_inv[V6] ≤ 1.0, littleleft[V10] ≤ 1.0, right[V10] ≤ 1.0, inv_left[V10]
≤ 1.0, right_pause[V10] ≤ 1.0, left[V10] ≤ 1.0, pause_right[V10] ≤ 1.0,
right_pause[V10] ≤ 1.0, up[V11] ≤ 1.0, down[V11] ≤ 1.0, pause_up[V11] ≤
1.0, pause_down[V11] ≤ 1.0, longdown[V11] ≤ 1.0, longup[V11] ≤ 1.0, inv_
longdown[V11] ≤ 1.0, behind[V12] ≤ 1.0, front[V12] ≤ 1.0, pause_front[V1
2] ≤ 1.0, front_pause[V12] ≤ 1.0, longfront[V12] ≤ 1.0, longbehind[V12]
≤ 1.0, pause_slightlybehind_front[V12] ≤ 1.0, ⟨A⟩left[V4] ≤ 1.0, ⟨A⟩righ
t[V4] ≤ 1.0, ⟨A⟩inv_left[V4] ≤ 1.0, ⟨A⟩pause_right[V4] ≤ 1.0, ⟨A⟩inv_lef
t_inv[V4] ≤ 1.0, ⟨A⟩pause_right_inv[V4] ≤ 1.0, ⟨A⟩left_inv_right[V4] ≤ 1
.0, ⟨A⟩down[V5] ≤ 1.0, ⟨A⟩up[V5] ≤ 1.0, ⟨A⟩down_pause[V5] ≤ 1.0, ⟨A⟩paus
e_up[V5] ≤ 1.0, ⟨A⟩longdown_pause[V5] ≤ 1.0, ⟨A⟩longup[V5] ≤ 1.0, ⟨A⟩lon
gpause_up[V5] ≤ 1.0, ⟨A⟩front[V6] ≤ 1.0, ⟨A⟩behind[V6] ≤ 1.0, ⟨A⟩behind_
pause[V6] ≤ 1.0, ⟨A⟩front_pause[V6] ≤ 1.0, ⟨A⟩longbehind[V6] ≤ 1.0, ⟨A⟩l
ongfront[V6] ≤ 1.0, ⟨A⟩front_inv[V6] ≤ 1.0, ⟨A⟩littleleft[V10] ≤ 1.0,
⟨A⟩right[V10] ≤ 1.0, ⟨A⟩inv_left[V10] ≤ 1.0, ⟨A⟩right_pause[V10] ≤ 1.0,
⟨A⟩left[V10] ≤ 1.0, ⟨A⟩pause_right[V10] ≤ 1.0, ⟨A⟩right_pause[V10] ≤ 1.0
, ⟨A⟩up[V11] ≤ 1.0, ⟨A⟩down[V11] ≤ 1.0, ⟨A⟩pause_up[V11] ≤ 1.0, ⟨A⟩pause
_down[V11] ≤ 1.0, ⟨A⟩longdown[V11] ≤ 1.0, ⟨A⟩longup[V11] ≤ 1.0, ⟨A⟩inv_l
ongdown[V11] ≤ 1.0, ⟨A⟩behind[V12] ≤ 1.0, ⟨A⟩front[V12] ≤ 1.0, ⟨A⟩pause_
front[V12] ≤ 1.0, ⟨A⟩front_pause[V12] ≤ 1.0, ⟨A⟩longfront[V12] ≤ 1.0,
⟨A⟩longbehind[V12] ≤ 1.0, ⟨A⟩pause_slightlybehind_front[V12] ≤ 1.0, ⟨B⟩l
eft[V4] ≤ 1.0, ⟨B⟩right[V4] ≤ 1.0, ⟨B⟩inv_left[V4] ≤ 1.0, ⟨B⟩pause_right
[V4] ≤ 1.0, ⟨B⟩inv_left_inv[V4] ≤ 1.0, ⟨B⟩pause_right_inv[V4] ≤ 1.0, ⟨B⟩
left_inv_right[V4] ≤ 1.0, ⟨B⟩down[V5] ≤ 1.0, ⟨B⟩up[V5] ≤ 1.0, ⟨B⟩down_pa
use[V5] ≤ 1.0, ⟨B⟩pause_up[V5] ≤ 1.0, ⟨B⟩longdown_pause[V5] ≤ 1.0, ⟨B⟩lo
ngup[V5] ≤ 1.0, ⟨B⟩longpause_up[V5] ≤ 1.0, ⟨B⟩front[V6] ≤ 1.0, ⟨B⟩behind
[V6] ≤ 1.0, ⟨B⟩behind_pause[V6] ≤ 1.0, ⟨B⟩front_pause[V6] ≤ 1.0, ⟨B⟩long
behind[V6] ≤ 1.0, ⟨B⟩longfront[V6] ≤ 1.0, ⟨B⟩front_inv[V6] ≤ 1.0, ⟨B⟩lit
tleleft[V10] ≤ 1.0, ⟨B⟩right[V10] ≤ 1.0, ⟨B⟩inv_left[V10] ≤ 1.0, ⟨B⟩righ
t_pause[V10] ≤ 1.0, ⟨B⟩left[V10] ≤ 1.0, ⟨B⟩pause_right[V10] ≤ 1.0, ⟨B⟩ri
ght_pause[V10] ≤ 1.0, ⟨B⟩up[V11] ≤ 1.0, ⟨B⟩down[V11] ≤ 1.0, ⟨B⟩pause_up
[V11] ≤ 1.0, ⟨B⟩pause_down[V11] ≤ 1.0, ⟨B⟩longdown[V11] ≤ 1.0, ⟨B⟩longup
[V11] ≤ 1.0, ⟨B⟩inv_longdown[V11] ≤ 1.0, ⟨B⟩behind[V12] ≤ 1.0, ⟨B⟩front
[V12] ≤ 1.0, ⟨B⟩pause_front[V12] ≤ 1.0, ⟨B⟩front_pause[V12] ≤ 1.0, ⟨B⟩lo
ngfront[V12] ≤ 1.0, ⟨B⟩longbehind[V12] ≤ 1.0, ⟨B⟩pause_slightlybehind_fr
ont[V12] ≤ 1.0, ⟨E⟩left[V4] ≤ 1.0, ⟨E⟩right[V4] ≤ 1.0, ⟨E⟩inv_left[V4] ≤
1.0, ⟨E⟩pause_right[V4] ≤ 1.0, ⟨E⟩inv_left_inv[V4] ≤ 1.0, ⟨E⟩pause_right
_inv[V4] ≤ 1.0, ⟨E⟩left_inv_right[V4] ≤ 1.0, ⟨E⟩down[V5] ≤ 1.0, ⟨E⟩up[V
5] ≤ 1.0, ⟨E⟩down_pause[V5] ≤ 1.0, ⟨E⟩pause_up[V5] ≤ 1.0, ⟨E⟩longdown_pa
```

```
use[V5] ≤ 1.0, ⟨E⟩longup[V5] ≤ 1.0, ⟨E⟩longpause_up[V5] ≤ 1.0, ⟨E⟩front
[V6] ≤ 1.0, ⟨E⟩behind[V6] ≤ 1.0, ⟨E⟩behind_pause[V6] ≤ 1.0, ⟨E⟩front_pau
se[V6] ≤ 1.0, ⟨E⟩longbehind[V6] ≤ 1.0, ⟨E⟩longfront[V6] ≤ 1.0, ⟨E⟩front_
inv[V6] ≤ 1.0, ⟨E⟩littleleft[V10] ≤ 1.0, ⟨E⟩right[V10] ≤ 1.0, ⟨E⟩inv_lef
t[V10] ≤ 1.0, ⟨E⟩right_pause[V10] ≤ 1.0, ⟨E⟩left[V10] ≤ 1.0, ⟨E⟩pause_ri
ght[V10] ≤ 1.0, ⟨E⟩right_pause[V10] ≤ 1.0, ⟨E⟩up[V11] ≤ 1.0, ⟨E⟩down[V1
1] ≤ 1.0, ⟨E⟩pause_up[V11] ≤ 1.0, ⟨E⟩pause_down[V11] ≤ 1.0, ⟨E⟩longdown
[V11] ≤ 1.0, ⟨E⟩longup[V11] ≤ 1.0, ⟨E⟩inv_longdown[V11] ≤ 1.0, ⟨E⟩behind
[V12] ≤ 1.0, ⟨E⟩front[V12] ≤ 1.0, ⟨E⟩pause_front[V12] ≤ 1.0, ⟨E⟩front_pa
use[V12] ≤ 1.0, ⟨E⟩longfront[V12] ≤ 1.0, ⟨E⟩longbehind[V12] ≤ 1.0, ⟨E⟩pa
use_slightlybehind_front[V12] ≤ 1.0, ⟨D⟩left[V4] ≤ 1.0, ⟨D⟩right[V4] ≤ 1
.0, ⟨D⟩inv_left[V4] ≤ 1.0, ⟨D⟩pause_right[V4] ≤ 1.0, ⟨D⟩inv_left_inv[V4]
≤ 1.0, ⟨D⟩pause_right_inv[V4] ≤ 1.0, ⟨D⟩left_inv_right[V4] ≤ 1.0, ⟨D⟩dow
n[V5] ≤ 1.0, ⟨D⟩up[V5] ≤ 1.0, ⟨D⟩down_pause[V5] ≤ 1.0, ⟨D⟩pause_up[V5] ≤
1.0, ⟨D⟩longdown_pause[V5] ≤ 1.0, ⟨D⟩longup[V5] ≤ 1.0, ⟨D⟩longpause_up[V
5] ≤ 1.0, ⟨D⟩front[V6] ≤ 1.0, ⟨D⟩behind[V6] ≤ 1.0, ⟨D⟩behind_pause[V6] ≤
1.0, ⟨D⟩front_pause[V6] ≤ 1.0, ⟨D⟩longbehind[V6] ≤ 1.0, ⟨D⟩longfront[V6]
≤ 1.0, ⟨D⟩front_inv[V6] ≤ 1.0, ⟨D⟩littleleft[V10] ≤ 1.0, ⟨D⟩right[V10] ≤
1.0, ⟨D⟩inv_left[V10] ≤ 1.0, ⟨D⟩right_pause[V10] ≤ 1.0, ⟨D⟩left[V10] ≤ 1
.0, ⟨D⟩pause_right[V10] ≤ 1.0, ⟨D⟩right_pause[V10] ≤ 1.0, ⟨D⟩up[V11] ≤ 1
.0, ⟨D⟩down[V11] ≤ 1.0, ⟨D⟩pause_up[V11] ≤ 1.0, ⟨D⟩pause_down[V11] ≤ 1.0
, ⟨D⟩longdown[V11] ≤ 1.0, ⟨D⟩longup[V11] ≤ 1.0, ⟨D⟩inv_longdown[V11] ≤ 1
.0, ⟨D⟩behind[V12] ≤ 1.0, ⟨D⟩front[V12] ≤ 1.0, ⟨D⟩pause_front[V12] ≤ 1.0
, ⟨D⟩front_pause[V12] ≤ 1.0, ⟨D⟩longfront[V12] ≤ 1.0, ⟨D⟩longbehind[V12]
≤ 1.0, ⟨D⟩pause_slightlybehind_front[V12] ≤ 1.0, ⟨O⟩left[V4] ≤ 1.0, ⟨O⟩r
ight[V4] ≤ 1.0, ⟨O⟩inv_left[V4] ≤ 1.0, ⟨O⟩pause_right[V4] ≤ 1.0, ⟨O⟩inv_
left_inv[V4] ≤ 1.0, ⟨O⟩pause_right_inv[V4] ≤ 1.0, ⟨O⟩left_inv_right[V4]
≤ 1.0, ⟨O⟩down[V5] ≤ 1.0, ⟨O⟩up[V5] ≤ 1.0, ⟨O⟩down_pause[V5] ≤ 1.0, ⟨O⟩p
ause_up[V5] ≤ 1.0, ⟨O⟩longdown_pause[V5] ≤ 1.0, ⟨O⟩longup[V5] ≤ 1.0, ⟨O⟩
longpause_up[V5] ≤ 1.0, ⟨O⟩front[V6] ≤ 1.0, ⟨O⟩behind[V6] ≤ 1.0, ⟨O⟩behi
nd_pause[V6] ≤ 1.0, ⟨O⟩front_pause[V6] ≤ 1.0, ⟨O⟩longbehind[V6] ≤ 1.0,
⟨O⟩longfront[V6] ≤ 1.0, ⟨O⟩front_inv[V6] ≤ 1.0, ⟨O⟩littleleft[V10] ≤ 1.0
, ⟨O⟩right[V10] ≤ 1.0, ⟨O⟩inv_left[V10] ≤ 1.0, ⟨O⟩right_pause[V10] ≤ 1.0
, ⟨O⟩left[V10] ≤ 1.0, ⟨O⟩pause_right[V10] ≤ 1.0, ⟨O⟩right_pause[V10] ≤ 1
.0, ⟨O⟩up[V11] ≤ 1.0, ⟨O⟩down[V11] ≤ 1.0, ⟨O⟩pause_up[V11] ≤ 1.0, ⟨O⟩pau
se_down[V11] ≤ 1.0, ⟨O⟩longdown[V11] ≤ 1.0, ⟨O⟩longup[V11] ≤ 1.0, ⟨O⟩inv
_longdown[V11] ≤ 1.0, ⟨O⟩behind[V12] ≤ 1.0, ⟨O⟩front[V12] ≤ 1.0, ⟨O⟩paus
e_front[V12] ≤ 1.0, ⟨O⟩front_pause[V12] ≤ 1.0, ⟨O⟩longfront[V12] ≤ 1.0,
⟨O⟩longbehind[V12] ≤ 1.0, ⟨O⟩pause_slightlybehind_front[V12] ≤ 1.0]

Items measures: [(ModalAssociationRules.gsupport, 0.1, 0.1)]
Rules measures: [(ModalAssociationRules.gconfidence, 0.5, 0.5)]

# of frequent patterns mined: 0
# of association rules mined: 0

Local measures memoization structure entries: 0
Global measures memoization structure entries: 0

Additional infos: [:size, :istrained]
Specialization fields: Symbol[]
```

In [34]:

```
mine!(miner)
```

Out[34]:
```
74-element Vector{ARule}:
 right[V4] ≤ 1.0 => ⟨O⟩inv_left[V4] ≤ 1.0
 down[V5] ≤ 1.0 => ⟨O⟩down_pause[V5] ≤ 1.0
 down[V5] ≤ 1.0 => ⟨O⟩littleleft[V10] ≤ 1.0
 up[V5] ≤ 1.0 => ⟨O⟩up[V5] ≤ 1.0
 up[V5] ≤ 1.0 => ⟨O⟩littleleft[V10] ≤ 1.0
 up[V5] ≤ 1.0 => ⟨O⟩up[V11] ≤ 1.0
 pause_up[V5] ≤ 1.0 => ⟨O⟩inv_left[V4] ≤ 1.0
 pause_up[V5] ≤ 1.0 => ⟨O⟩up[V5] ≤ 1.0
 pause_up[V5] ≤ 1.0 => ⟨O⟩front[V6] ≤ 1.0
 pause_up[V5] ≤ 1.0 => ⟨O⟩right_pause[V10] ≤ 1.0
 pause_up[V5] ≤ 1.0 => ⟨O⟩up[V11] ≤ 1.0
 pause_up[V5] ≤ 1.0 => ⟨O⟩pause_up[V11] ≤ 1.0
 longdown_pause[V5] ≤ 1.0 => ⟨D⟩inv_left[V4] ≤ 1.0
 ⋮
 longup[V11] ≤ 1.0 => ⟨O⟩pause_down[V11] ≤ 1.0
 longup[V11] ≤ 1.0 => ⟨O⟩longup[V11] ≤ 1.0
 longup[V11] ≤ 1.0 => ⟨O⟩inv_longdown[V11] ≤ 1.0
 behind[V12] ≤ 1.0 => ⟨O⟩littleleft[V10] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩down[V5] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩down_pause[V5] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩behind[V6] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩front_pause[V6] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩inv_left[V10] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩down[V11] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩pause_down[V11] ≤ 1.0
 longfront[V12] ≤ 1.0 => ⟨D⟩front[V12] ≤ 1.0
```

In [35]:
```
length(freqitems(miner))
```

Out[35]:
```
370
```

In [36]:
```
length(arules(miner))
```

Out[36]:
```
74
```

In [37]:
```
arules(miner)
```

```
Out[37]: 74-element Vector{ARule}:
 right[V4] ≤ 1.0 => ⟨O⟩inv_left[V4] ≤ 1.0
 down[V5] ≤ 1.0 => ⟨O⟩down_pause[V5] ≤ 1.0
 down[V5] ≤ 1.0 => ⟨O⟩littleleft[V10] ≤ 1.0
 up[V5] ≤ 1.0 => ⟨O⟩up[V5] ≤ 1.0
 up[V5] ≤ 1.0 => ⟨O⟩littleleft[V10] ≤ 1.0
 up[V5] ≤ 1.0 => ⟨O⟩up[V11] ≤ 1.0
 pause_up[V5] ≤ 1.0 => ⟨O⟩inv_left[V4] ≤ 1.0
 pause_up[V5] ≤ 1.0 => ⟨O⟩up[V5] ≤ 1.0
 pause_up[V5] ≤ 1.0 => ⟨O⟩front[V6] ≤ 1.0
 pause_up[V5] ≤ 1.0 => ⟨O⟩right_pause[V10] ≤ 1.0
 pause_up[V5] ≤ 1.0 => ⟨O⟩up[V11] ≤ 1.0
 pause_up[V5] ≤ 1.0 => ⟨O⟩pause_up[V11] ≤ 1.0
 longdown_pause[V5] ≤ 1.0 => ⟨D⟩inv_left[V4] ≤ 1.0
 ⋮
 longup[V11] ≤ 1.0 => ⟨O⟩pause_down[V11] ≤ 1.0
 longup[V11] ≤ 1.0 => ⟨O⟩longup[V11] ≤ 1.0
 longup[V11] ≤ 1.0 => ⟨O⟩inv_longdown[V11] ≤ 1.0
 behind[V12] ≤ 1.0 => ⟨O⟩littleleft[V10] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩down[V5] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩down_pause[V5] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩behind[V6] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩front_pause[V6] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩inv_left[V10] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩down[V11] ≤ 1.0
 front_pause[V12] ≤ 1.0 => ⟨O⟩pause_down[V11] ≤ 1.0
 longfront[V12] ≤ 1.0 => ⟨D⟩front[V12] ≤ 1.0
```