

## A Map File

The following is the input file to generate the program mount using the benchmark generating tool.

```

states: 10
average_cities_per_state: 15
standard_deviation: 3
MONTANA {(0,2500);(2000,3500)}
WYOMING {(500,1500);(2000,2500)}
COLORADO {(700,500);(2200,1500)}
NORTH_DAKOTA {(2000,2800);(3200,3500)}
SOUTH_DAKOTA {(2000,2100);(3200,2800)}
NEBRASKA {(2000,1500);(3200,2100)}
KANSAS {(2200,500);(3500,1500)}
MINNESOTA {(3200,2500);(3700,3500)}
IOWA {(3200,1500);(4000,2500)}
MISSOURI {(3500,0);(4100,1500)}
MONTANA NORTH_DAKOTA
NORTH_DAKOTA MINNESOTA
MINNESOTA IOWA
IOWA MISSOURI
MISSOURI KANSAS
KANSAS COLORADO
COLORADO WYOMING
WYOMING NEBRASKA
NEBRASKA SOUTH_DAKOTA
SOUTH_DAKOTA MONTANA
end

```

## B WME templates

Following is a list of templates to create WMEs for each state generated by the tool. Also you will find the templates used for pairs of states. In the code below the characters @, #, and % are used as wild cards. They are automatically replaced by the name of the states in the map provided by the user.

```

; *****
; WMEs used for a specific state
;
(literalize @_city status name fitness)
(literalize @_distance from to value)

```

```

(literalize @_distance_to_border city
state value)
(literalize @_trip status from to distance)
(literalize @_next_city name fitness)
(literalize @_optimization status city_1
city_2 fitness)
(literalize @_border_city name)
(literalize @_all_borders_dist_computed
value)
; *****
; WMEs used for pair of states
;
(literalize @_#_border status)
(literalize @_#_crossing from_status
from_city to_status to_city
distance)
(literalize @_border_distance from to value)
(literalize @_border_trip status position
from to distance)
(literalize @_#_distance from to value)
(literalize @_#_optim_from from_city to_city
fitness)
(literalize @_#_optim_to from_city to_city
fitness)
(literalize @_city status name fitness)
(literalize @_distance from to value)
(literalize @_distance_to_border city state
value)
(literalize @_trip status from to distance)
(literalize @_next_city name fitness)
(literalize @_optimization status city_1
city_2 fitness)
(literalize @_border_city name)
(literalize @_all_borders_dist_computed
value)

```

## C Rule Templates

Next we present the actual rule templates used by the benchmark generating tool to create the programs. In these rules also the wild cards get replaced by state names.

```

; *****
; We are done when all the inner trip
; writings are done
(@_city ^name <c2>
^status first_city_candidate)
-(@_city ^status in_trip)
-->
(modify 2 ^status not_in_trip))

(p termination
(inner_trips_writing ^status done)
-(inner_trips_writing ^status undone)
-(inner_trips_writing ^status finito)
-->
(write end)
(modify 1 ^status finito))

; *****
; *****
(p @_put_first_city_in_trip
(@_city ^name <> ^status first_city)
-(@_city ^status first_city_candidate)
-->
(modify 1 ^status in_trip))

; *****
; *****
(p @_create_first_city_candidates
(@_all_borders_dist_computed ^value true)
(@_city ^name <c1> ^status not_in_trip)
(%_@_border)
(@_distance_to_border ^city <c1>
^state % ^value <df>)
(@_#_border)
(@_distance_to_border ^city <c1>
^state # ^value <dt>)
-(@_city ^status in_trip)
-(@_city ^status first_city)
-->
(modify 2 ^status first_city_candidate
^fitness (compute (<df> - <dt>))))

; *****
; *****
(p @_create_border_trips_1
(@_trip ^from <c1> ^to <c2> ^distance <d12>)
(@_trip ^from <c2> ^to <c3> ^distance <d23>)
(@_distance ^from <c1> ^to <c3>
^value <d13>)
-(@_city ^status not_in_trip)
-(@_city ^status first_city)
-(@_city ^status just_chosen)
-(@_city ^status first_city_candidate)
-(@_city ^status next_city_candidate)
-(@_optimization)
-(@_trip ^to <c1>)
-->
(make @_border_trip ^status created
^position begin ^from <c1> ^to <c2>
^distance <d12>)
(make @_border_trip ^status created
^position begin ^from <c2> ^to <c3>
^distance <d23>)
(make @_border_distance ^from <c1>
^to <c2> ^value <d12>)
(make @_border_distance ^from <c2>
^to <c3> ^value <d23>)
(make @_border_distance ^from <c1>
^to <c3> ^value <d13>)
(make @_border_distance ^from <c2>
^to <c1> ^value <d12>)
(make @_border_distance ^from <c3>
^to <c2> ^value <d23>))

(p @_choose_first_city
(@_city ^name <c1>
^status first_city_candidate ^fitness <f>)
-(@_city ^status first_city_candidate
^fitness < <f>)
-(@_city ^status not_in_trip)
-(@_city ^status first_city)
-->
(modify 1 ^status first_city))

; *****
; *****
(p @_remove_first_city_candidates
(@_city ^name <c1> ^status first_city)

```

```

(make @_border_distance ^from <c3>
      ^to <c1> ^value <d13>)
(make @_border_city ^name <c1>)
(make @_border_city ^name <c2>)
(make @_border_city ^name <c3>))

; *****

(p @_create_border_trips_2
  (@_trip ^from <c1> ^to <c2> ^distance <d12>)
  (@_trip ^from <c2> ^to <c3> ^distance <d23>)
  (@_distance ^from <c1> ^to <c3>
    ^value <d13>)
  -(@_city ^status not_in_trip)
  -(@_city ^status first_city)
  -(@_city ^status just_chosen)
  -(@_city ^status first_city_candidate)
  -(@_city ^status next_city_candidate)
  -(@_optimization)
  -(@_trip ^from <c3>)
  -->
  (make @_border_trip ^status created
    ^position end ^from <c1> ^to <c2>
      ^distance <d12>); *****
  (make @_border_trip ^status created
    ^position end ^from <c2> ^to <c3>
      ^distance <d23>)
  (make @_border_distance ^from <c1>
    ^to <c2> ^value <d12>)
  (make @_border_distance ^from <c2>
    ^to <c3> ^value <d23>)
  (make @_border_distance ^from <c1>
    ^to <c3> ^value <d13>)
  (make @_border_distance ^from <c2>
    ^to <c1> ^value <d12>)
  (make @_border_distance ^from <c3>
    ^to <c2> ^value <d23>)
  (make @_border_distance ^from <c3>
    ^to <c1> ^value <d13>)
  (make @_border_city ^name <c1>)
  (make @_border_city ^name <c2>)
  (make @_border_city ^name <c3>))

(p @_make_next_city_candidates
  (@_city ^name <c2> ^status in_trip)
  (@_city ^name <c3> ^status not_in_trip)
  (@_distance ^from <c2> ^to <c3>
    ^value <d23>)
  (@_#_border)
  (@_distance_to_border ^city <c3>
    ^state # ^value <d_c3_s>)
  -(@_city ^status first_city_candidate)
  -(@_city ^status just_chosen)
  -(@_trip ^from <c2>)
  -->
  (modify 2 ^status next_city_candidate
    ^fitness (compute (<d23> - <d_c3_s>))))

(p @_choose_next_city
  (@_city ^name <c2> ^status in_trip)
  (@_city ^name <c3>
    ^status next_city_candidate ^fitness <f>)
  (@_distance ^from <c2> ^to <c3>
    ^value <d23>)
  -(@_city ^status next_city_candidate
    ^fitness < <f>)
  -(@_trip ^from <c2>)
  -(@_city ^status not_in_trip)
  -(@_city ^status first_city_candidate)
  -(@_city ^status just_chosen)
  -->
  (make @_trip ^status not_printed
    ^from <c2> ^to <c3> ^distance <d23>)
  (modify 2 ^status just_chosen))

; *****

(p @_discharge_remaining_candidates
  (@_city ^name <c1> ^status just_chosen)

```

The criterion to select the next city in the tour involves selecting the closest city and also the one

that is farthest from the final border. It is desirable to choose the cities closer to the final border at the end of the tour. The heuristic compares the difference between the distances of a newly selected city to the last city included in the tour and the final border in the current state.



```

        ^to <c2> ^distance <d23>)
(make @_trip ^status not_printed ^from <c2>
 ^to <c4> ^distance <d24>))
; *****
(p @_remove_implemented_optimization
 (@_optimization ^status implemented
 ^city_1 <c2> ^city_2 <c3>)
 -(@_optimization ^status generated)
 -->
 (remove 1))
; *****
(@_border_trip ^status modified ^from <ca>
 ^to <cc> ^distance <d32>)
-->
(remove 1)
(remove 2)
(modify 3 ^status corrected)
(modify 4 ^status corrected)
(make @_trip ^status not_printed ^from <cb>
 ^to <ca> ^distance <d13>)
(make @_trip ^status not_printed ^from <ca>
 ^to <cc> ^distance <d32>))
; *****
The rules that optimize the border are independent
of the rules that generate the internal tour within
the state. At times there might be some conflicts
between the internal trips of one state and the border
trips. Whenever this happens, the border trips are
maintained.
(p @_correct_conflicting_trips_1
 (@_trip ^from <c1> ^to <c2>)
 (@_trip ^from <c2> ^to <c3>)
 (@_border_trip ^status modified ^from <c1>
 ^to <c3> ^distance <d13>)
 (@_border_trip ^status modified ^from <c3>
 ^to <c2> ^distance <d32>)
 -->
 (remove 1)
 (remove 2)
 (modify 3 ^status corrected)
 (modify 4 ^status corrected)
 (make @_trip ^status not_printed ^from <c1>
 ^to <c3> ^distance <d13>)
 (make @_trip ^status not_printed ^from <c3>
 ^to <c2> ^distance <d32>))
; *****
; *****
(p @_final_print
 (@_#_border ^status printed)
 (%_@_border ^status printed)
 (@_trip ^status not_printed ^from <c1>
 ^to <c2>))
 -(@_border_trip ^status modified)
 -->
 (write R13: <c1> <c2>)
 (modify 3 ^status printed))
; *****
(p @_printing_done
 (inner_trips_writing ^status undone
 ^state @)
 (@_#_border ^status printed)
 (%_@_border ^status printed)
 -(@_trip ^status not_printed)
 -->
 (modify 1 ^status done))
; *****
; Make border crossing
(p @_#_make_crossing
 (@_#_border ^status undone)
 (#_border_trip ^position begin ^from <ca>
 ^to <cb> ^distance <dab>)
 (@_border_trip ^position end ^from <c2>
 ^to <c3> ^distance <d23>)
 (@_#_distance ^from <c3> ^to <ca>
 ^value <d3a>))
; *****
(p @_correct_conflicting_trips_2
 (@_trip ^from <ca> ^to <cb>)
 (@_trip ^from <cb> ^to <cc>)
 (@_border_trip ^status modified ^from <cb>
 ^to <ca> ^distance <d13>))

```

```

- (#_border_trip ^to <ca>) (compute (<dbc>+(<d3a>-(<d3b>+<dac>))))))
- (@_border_trip ^from <c3>)
--> ; *****
(modify 1 ^status done) ;
(make @_#_crossing ^from_status generated
  ^to_status generated ^from_city <c3> (p @_#_eliminate_negative_optim_from
  ^to_city <ca> ^distance <d3a>)) (@_#_optim_from ^fitness {<f> < 0})
-->
; ***** (remove 1)
;
(p @_#_make_optim_candidate_from (p @_#_eliminate_negative_optim_to
  (@_#_crossing ^from_status generated (@_#_optim_to ^fitness {<f> < 0})
  ^from_city <c3> ^to_city <ca> -->
  ^distance <d3a>) (remove 1))
  (@_border_trip ^from <c2> ^to <c3>) ; *****
  (@_border_trip ^from <c1> ^to <c2> ;
  ^distance <d12>))
  (@_border_distance ^from <c1> ^to <c3> (p @_#_optimize_connection_from
  ^value <d13>)) (@_#_crossing ^from_city <c3>
  (@_#_distance ^from <c2> ^to <ca> ^to_city <ca>)
  ^value <d2a>)) (@_#_optim_from ^from_city <c3>
  ^to_city <ca> ^fitness {<f> > 0})
--> (@_border_trip ^from <c2>
  (modify 1 ^from_status optimized) ^to <c3> ^distance <d23>))
  (make @_#_optim_from ^from_city <c3> (@_border_trip ^from <c1> ^to <c2>))
  ^to_city <ca> ^fitness (@_border_distance ^from <c1> ^to <c3>
  (compute (<d12>+(<d3a>-(<d13>+<d2a>)))))) ^value <d13>))
; ***** (@_#_distance ^from <c2> ^to <ca>
; ^value <d2a>))
- (@_#_optim_to ^from_city <c3>
  ^to_city <ca> ^fitness > <f>))
(p @_#_make_optim_candidate_to -->
  (@_#_crossing ^to_status generated (remove 1)
  ^from_city <c3> ^to_city <ca> (remove 2)
  ^distance <d3a>) (remove 3)
  (#_border_trip ^from <ca> ^to <cb>) (remove 4)
  (#_border_trip ^from <cb> ^to <cc>
  ^distance <dbc>) (make @_border_trip ^status modified
  ^from <c1> ^to <c3> ^distance <d13>))
  (#_border_distance ^from <ca> ^to <cc> ^value <dac>) (make @_border_trip ^status modified
  ^from <c3> ^to <c2> ^distance <d23>))
  (@_#_distance ^from <c3> ^to <cb>
  ^value <d3b>) (make @_#_crossing ^from_status optimized
  ^to_status generated ^from_city <c2>
  ^to_city <ca> ^distance <d2a>))
-->
(modify 1 ^to_status optimized)
(make @_#_optim_to ^from_city <c3>
  ^to_city <ca> ^fitness ; *****

```

```

;
(p @#_remove_invalid_optim_from
  (@#_optim_from ^from_city <c3>
    ^to_city <ca>)
  -(@#_crossing ^from_city <c3>
    ^to_city <ca>)
  -->
  (remove 1))

(p @#_remove_invalid_optim_to
  (@#_optim_to ^from_city <c3>
    ^to_city <ca>)
  -(@#_crossing ^from_city <c3>
    ^to_city <ca>)
  -->
  (remove 1))

; *****

(p @#_optimize_border_to
  (@#_crossing ^from_city <c3>
    ^to_city <ca>)
  (@#_optim_to ^from_city <c3>
    ^to_city <ca> ^fitness {<f> > 0})
  (#_border_trip ^from <ca> ^to <cb>
    ^distance <dab>)
  (#_border_trip ^from <cb> ^to <cc>)
  (#_border_distance ^from <ca> ^to <cc>
    ^value <dac>)
  (@#_distance ^from <c3> ^to <cb>
    ^value <d3b>)
  -(@#_optim_from ^from_city <c3>
    ^to_city <ca> ^fitness > <f>)
  -->
  (remove 1)
  (remove 2)
  (remove 3)
  (remove 4)
  (make #_border_trip ^status modified
    ^from <cb> ^to <ca> ^distance <dab>)
  (make #_border_trip ^status modified
    ^from <ca> ^to <cc> ^distance <dac>)
  (make @#_crossing ^from_status generated
    ^to_status optimized ^from_city <c3>
    ^to_city <cb> ^distance <d3b>))

```