Retrieve the new listings in El Segundo every day:

(a, Retrieve(new-listings,MLS), b)

(b, Store(new-listings,old-listings), c)

(c, Report(new-listings,email), d)

(d, Sleep(1 day), a)

Report the weather forecasts each month

(a, Retrieve(forecast, CNN), b)

(b, Store(forecast, old-weather), c)

(c, Retrieve(date,Unix), d)

(d, Test

Given the following sources:

Real Estate Site:

real\_estate(mls#,street-address,city,price,bedrooms,baths,photo)

local\_data(mls#)

Plan structure:

({x}, action, {y})

x -- enabled in initial state

If x is enabled, disable x, execute action

if action executes sucessfully, enable y

Exit when no actions are enabled

When more than one action is enabled, execute in parallel (or either order)

{x} can also be a set

Actions:

retrieve q,d - retrieve the set of data through Ariadne for query q producing data d

select d,c,ds - select from data d, using the constraint c producing data ds

project d,a,dp - project from data d attributes a producing data dp

join d1,d2,jc,d3 - join d1 and d2 over jc producing d3

union d1,d2,d3 - union d1 and d2 producing d3

set-diff d1,d2,d3 - take the set difference of d1-d2 producing d3

aggregate d,f,ad - aggregate data d, using function f, producing ad

store r, d - add to the set of local data (must already exist)

create r,d - store the data as a new relation r

delete r - remove the set of locally stored data

sleep x y - create an at process to restart in time x at step y, exit

format d,t,dt - format d using template t producing dt

notify a,t,d - notify at address a, type t, data d

e.g., type = email, address = email address, date = d

type = phone, address = number, data = voice(d)

type = fax, address = number, data = d

type = popup window, address = window loc, data = d

test a,oc,b - test order constraint oc on unary data a and b

(a, retrieve(real\_estate(?mls#)), b)

(b,