```
/*
 * THE GUIDE TO NONEXISTENT BIRDS - AN ORNITHOLOGICAL LOGIC
 * Edited by Kavi Duvvoori
 *
 *
 *
 * (How I would recommend reading this thing, though there is room for
 * debate on this question - read the Neruda poem, all the comments I
 * wrote (I’m attached to a few of them), the Wallace Stevens in its
 * entirety but only once, skip the Keats except for a few lines for
 * atmosphere, the Bukowski won’t take up too much of your time (the
 * poems are included not because i think they’re the very best bird
 * related poems there are but because maybe they somehow help develop
 * the narrative), skim 13 or so birds, and don’t bother with the code
 * except for a few carefully named variables; it’s up to you what to do
 * with the rest of it)
 *
 *-- Returning 5 years later: I was 19. It was my first try at digital
 *-- lit. Some I excuse as the work of "trying on thoughts," some I don’t.
 *--
 *-- I believe I understood the faint irony of using these poets but
 *-- failed to work with their cliche. Only Stevens is faintly defensible.
 *-- But none of that matters, not in the scheme of things, we all know.
 *--
 *-- I wish I could not have included the following poem, because Pablo
 *-- Neruda was a rapist. He wrote it lyrically in his autobiography or he
 *-- might have died with his rape, but no one really noticed, for decades
 *-- it seems. You see, he wrote his rape (of a Sri Lankan woman, where he
 *-- was ambassador) in the way he wrote his poetry, unrepentant except to
 *-- turn the mood as a shift of register. She was in a position, by
 *-- language, class & caste, asymmetry of power, incapable, literally,
 *-- of speech as a subject. And then, towards the end of his life, he
 *-- wrote again his rape (after no one asked) in a way that acknowledges
 *-- irrelevant shame but reenacts his, her society’s, literary criticism’s
 *-- denial of her, who I do assume real, humanity. I heard about it in my
 *-- Spanish class and, weirdly for its commonality, couldn’t read that
 *-- week (I admit this is a performative morality).
 *--
 *-- The poem came after the project. I was not intensely or particularly
 *-- a fan. But it was the schematic of my essay (this): it was talking
 *-- about the (literal) content I was talking about, in a totally
 *-- different rhetoric, well or badly effectively irrelevant. I will
 *-- write about sexual violence here not because I have yet done the work
 *-- to be a person to write about it; but I acknowledge I was writing it
 *-- already, so if I intend to keeping using this essay: more, using
 *-- this essay as the messy incipience of a certain intellectual project
 *-- (mine), a portfolio piece. That silence cannot be reinscribed.
 *
 * BIRD
 * Pablo Neruda
 *
 * It was passed from one bird to another,
 * the whole gift of the day.
 * The day went from flute to flute,
 * went dressed in vegetation,
 * in flights which opened a tunnel
 * through which the wind would pass
 * to where birds were breaking open
 * the dense blue air -
 * and there, night came in.
 *
 * When I returned from so many journeys,
 * I stayed suspended and green
 * between sun and geography -
 * I saw how wings worked,
 * how perfumes are transmitted
 * by feathery telegraph,
 * and from above I saw the path,
 * the springs and the roof tiles,
 * the fishermen at their trades,
 * the trousers of the foam;
 * I saw it all from my green sky.
 * I had no more alphabet
 * than the swallows in their courses,
 * the tiny, shining water
 * of the small bird on fire
 * which dances out of the pollen
 *
 */

:- use_module(library(http/http_unix_daemon)).
:- use_module(library(http/thread_httpd)).
:- use_module(library(http/http_dispatch)).
:- use_module(library(http/http_error)).
:- use_module(library(http/html_write)).

:- initialization http_daemon.

% NOTE: We cannot yet guarantee that any birds described in this guide
% do not, in fact, exist. We deeply and sincerely apologize for any
% inconvenience this causes and, pending venture capitalist funding,
% are working on developing a solution to this problem.

fibonacci(0,0).
fibonacci(1,1).
fibonacci(N, FIB_N):-
        N >= 2,
        NMONE is N-1,
        NMTWO is N-2,
        fibonacci(NMONE,FIB_NMONE),
        fibonacci(NMTWO,FIB_NMTWO),
        FIB_N is FIB_NMONE + FIB_NMTWO.

strs_flatten(STR_LIST,FLAT_STR) :-
        maplist(string_chars, STR_LIST, CHAR_LISTS),
        flatten(CHAR_LISTS, FLAT_CHARS),
        string_chars(FLAT_STR,FLAT_CHARS).

capitalize_first("","").
capitalize_first(STR,STR_START_CAPITALIZED):-
        string_chars(STR,CHARS),
        CHARS = [H|R],
        upcase_atom(H,CH),
        NEW_CHARS = [CH|R],
        string_chars(STR_START_CAPITALIZED,NEW_CHARS).

server :-
    server(8080).
server(Port) :-
    http_server(http_dispatch, [port(Port)]).

:- http_handler('/ornithological/', say_birds, []).
:- http_handler('/ornithological', say_birds, []).

run :-
    server,

/*
 * THIRTEEN WAYS OF LOOKING AT A BLACKBIRD
 * Wallace Stevens
 *
 * I
 * Among twenty snowy mountains,
 * The only moving thing
 * Was the eye of the blackbird.
 * ...
 */

% We will start with a few of the names necessary for any bird watcher,
% or bird-imaginer


birdFamilies(["hummingbird","thrush","tinamou","egret","hawk","kestrel",
              "eagle","duck","falcon","partridge","brush-turkey","grebe",
              "coot","swallow","grouse","guineafowl","woodpecker",
              "shellduck","barbet","vulture","gull","flycatcher","swift",
              "albatross","oriole"]).

/*
 * THIRTEEN WAYS OF LOOKING AT A BLACKBIRD
 * Wallace Stevens
 *
 * I
 * Among twenty snowy mountains,
 * The only moving thing
 * Was the eye of the blackbird.
 * ...
 */

% And start naming them - maybe pointing in a room of taxidermied
% coots and grebes, eagles and egrets watching you from
% their paper-mached perches.

colors(["brown","blue","red","golden","crimson","white","black","gray",
        "yellow","violet"]).
birdParts(["tail","wing","head","beak","throat","shoulder","breast",
           "crest","neck"]).
climes(["tropical","arctic","coastal","prairie","river","forest",
        "mountain"]).
cardDirs(["north","south","east","west"]).
rarities(["common","uncommon","rare"]).
descTypes([coloredPart,clime,cardDir,rarity]).
partFeatures(["speckled","mottled","tufted","striped","narrow","large",
              "bright"]).
birdDiets([omnivorous,pescatarian,carnivorous,nectar]).
waysOfSinging(["silence","squawking","singing","chattering","tweeting",
               "screeching","whistling","groaning"]).

birdFamily_diet(FAMILY,omnivorous):-
        FAMILY = "thrush";
        FAMILY = "swallow";
        FAMILY = "swift";
        FAMILY = "flycatcher";
        FAMILY = "barbet";
        FAMILY = "tinamou";
        FAMILY = "partridge";
        FAMILY = "grouse";
        FAMILY = "brush-turkey";
        FAMILY = "guineafowl";
        FAMILY = "oriole";
        FAMILY = "woodpecker".

birdFamily_diet(FAMILY,pescatarian):-
        FAMILY = "egret";
        FAMILY = "grebe";
        FAMILY = "coot";
        FAMILY = "duck";
        FAMILY = "shellduck";
        FAMILY = "gull";
        FAMILY = "albatross".

birdFamily_diet(FAMILY,carnivorous):-
        FAMILY = "hawk";
        FAMILY = "eagle";
        FAMILY = "falcon";
        FAMILY = "kestrel";
        FAMILY = "vulture".

birdFamily_diet(FAMILY,nectar):-
        FAMILY = "hummingbird".

climePlaces(["swamps","undergrowth","canopies","rivers","tundra","mountain tops",
             "conifers","estuaries","beaches","bushes","treetops","shrubbery",
             "lakes","reeds","ponds","branches","cliffsides","pine stands"]).

clime_climePlaces("tropical",["swamps","undergrowth","canopies","rivers"]).
clime_climePlaces("arctic",["tundra","mountain tops","conifers"]).
clime_climePlaces("coastal",["estuaries","beaches","undergrowth","bushes"]).
clime_climePlaces("prairie",["treetops","shrubbery","lakes"]).
clime_climePlaces("river",["reeds","ponds","bushes","rivers","lakes"]).
clime_climePlaces("forest",["treetops","undergrowth","bushes","branches"]).
clime_climePlaces("mountain",["cliffsides","pine stands","bushes"]).

diet_habitat_foodSource(omnivorous,"swamps","insects and tubers").
diet_habitat_foodSource(omnivorous,"undergrowth","seeds, worms, and berries").
diet_habitat_foodSource(omnivorous,"canopies","nuts, beetles, and fruit").
diet_habitat_foodSource(omnivorous,"rivers","seeds, berries, and snails").
diet_habitat_foodSource(omnivorous,"tundra","roots and seeds").
diet_habitat_foodSource(omnivorous,"mountain tops","seeds and nuts").
diet_habitat_foodSource(omnivorous,"conifers","nuts, seeds, and beetles").
diet_habitat_foodSource(omnivorous,"estuaries","seeds and some grubs").
diet_habitat_foodSource(omnivorous,"beaches",
                        "seeds and driftwood-feeding insects").
diet_habitat_foodSource(omnivorous,"bushes","seeds, grubs, and berries").
diet_habitat_foodSource(omnivorous,"treetops","fruits and nuts").
diet_habitat_foodSource(omnivorous,"shrubbery","seeds and grubs").
diet_habitat_foodSource(omnivorous,"lakes","berries and aquatic insects").
diet_habitat_foodSource(omnivorous,"reeds","grubs and worms").
diet_habitat_foodSource(omnivorous,"ponds","insects").
diet_habitat_foodSource(omnivorous,"branches","fruits and nuts").
diet_habitat_foodSource(omnivorous,"cliffsides","seeds and spiders").
diet_habitat_foodSource(omnivorous,"pine stands","pine cones and small insects").

diet_habitat_foodSource(carnivorous,"swamps","amphibians and fish").
diet_habitat_foodSource(carnivorous,"undergrowth","rodents").
diet_habitat_foodSource(carnivorous,"canopies","small birds").
diet_habitat_foodSource(carnivorous,"rivers","fresh-water fish").
diet_habitat_foodSource(carnivorous,"tundra",
                        "lemmings, squirrels, and arctic rabbits").
diet_habitat_foodSource(carnivorous,"mountain tops","rabbits and small birds").
diet_habitat_foodSource(carnivorous,"conifers","rodents and wrens").
diet_habitat_foodSource(carnivorous,"estuaries","frogs and fish").
diet_habitat_foodSource(carnivorous,"beaches","shorebirds").
diet_habitat_foodSource(carnivorous,"bushes","rabbits, mice, and gophers").
diet_habitat_foodSource(carnivorous,"treetops",
                        "squirrels, mice, and other birds").
diet_habitat_foodSource(carnivorous,"shrubbery","rabbits and mice").
diet_habitat_foodSource(carnivorous,"lakes","fish and frugs").
diet_habitat_foodSource(carnivorous,"reeds","rodents and lizards").
diet_habitat_foodSource(carnivorous,"ponds","fish, frogs, rodents, and lizards").
diet_habitat_foodSource(carnivorous,"branches","squirrels").
diet_habitat_foodSource(carnivorous,"cliffsides","other birds").
diet_habitat_foodSource(carnivorous,"pine stands","flocks of small birds").

diet_habitat_foodSource(pescatarian,"swamps",
                        "catfish, bass, sunfish, and minnows").
diet_habitat_foodSource(pescatarian,"undergrowth","freshwater fish").
diet_habitat_foodSource(pescatarian,"canopies","trout, flounder, and perch").
diet_habitat_foodSource(pescatarian,"rivers","trout and other freshwater fish").
diet_habitat_foodSource(pescatarian,"tundra","salmon and mackerel").
diet_habitat_foodSource(pescatarian,"mountain tops",
                        "brittlefish, lanternfish, and eelouts").
diet_habitat_foodSource(pescatarian,"conifers","trout, salamander, and snails").
diet_habitat_foodSource(pescatarian,"estuaries","smelt, bass, and snappers").
diet_habitat_foodSource(pescatarian,"beaches",
                        "crabs, mussels, and a variety of saltwater fish").
diet_habitat_foodSource(pescatarian,"bushes","freshwater fish").
diet_habitat_foodSource(pescatarian,"treetops","freshwater fish").
diet_habitat_foodSource(pescatarian,"shrubbery","freshwater fish").
diet_habitat_foodSource(pescatarian,"lakes","berries and aquatic insects").
diet_habitat_foodSource(pescatarian,"reeds","carp, bream, trout, and perch").
diet_habitat_foodSource(pescatarian,"ponds","catfish, bass, and carp").
diet_habitat_foodSource(pescatarian,"branches","freshwater fish").
diet_habitat_foodSource(pescatarian,"cliffsides","rockfish and skates").
diet_habitat_foodSource(pescatarian,"pine stands","freshwater fish").

diet_habitat_foodSource(nectar,_,"the nectar of wildflowers").
```

```
/*
 * THIRTEEN WAYS OF LOOKING AT A BLACKBIRD
 * Wallace Stevens
 *
 * I
 * Among twenty snowy mountains,
 * The only moving thing
 * Was the eye of the blackbird.
 *
 * II
 * I was of three minds,
 * Like a tree
 * In which there are three blackbirds
 * ...
 */

% But, standing in rain-pants in some swamp, hopeful binoculars held
% chest high, what wings will you spot silhouetted, under
% bush or over roof?

birdFamily_seed("hummingbird",SEED):- SEED =< 0.01.
birdFamily_seed("thrush",SEED):- SEED >= 0.01, SEED < 0.08.
birdFamily_seed("tinamou",SEED):- SEED >= 0.08, SEED < 0.1.
birdFamily_seed("egret",SEED):- SEED >= 0.1, SEED < 0.15.
birdFamily_seed("oriole",SEED):- SEED >= 0.15, SEED < 0.2.
birdFamily_seed("hawk",SEED):- SEED >= 0.2, SEED < 0.28.
birdFamily_seed("kestrel",SEED):- SEED >= 0.28, SEED < 0.3.
birdFamily_seed("eagle",SEED):- SEED >= 0.3, SEED < 0.35.
birdFamily_seed("duck",SEED):- SEED >= 0.35, SEED < 0.41.
birdFamily_seed("falcon",SEED):- SEED >= 0.41, SEED < 0.42.
birdFamily_seed("partridge",SEED):- SEED >= 0.42, SEED < 0.45.
birdFamily_seed("brush-turkey",SEED):- SEED >= 0.45, SEED < 0.46.
birdFamily_seed("grebe",SEED):- SEED >= 0.46, SEED < 0.5.
birdFamily_seed("coot",SEED):- SEED >= 0.5, SEED < 0.55.
birdFamily_seed("swallow",SEED):- SEED >= 0.55, SEED < 0.63.
birdFamily_seed("grouse",SEED):- SEED >= 0.63, SEED < 0.66.
birdFamily_seed("guineafowl",SEED):- SEED >= 0.66, SEED < 0.69.
birdFamily_seed("shellduck",SEED):- SEED >= 0.69, SEED < 0.71.
birdFamily_seed("woodpecker",SEED):- SEED >= 0.71, SEED < 0.75.
birdFamily_seed("barbet",SEED):- SEED >= 0.75, SEED < 0.76.
birdFamily_seed("vulture",SEED):- SEED >= 0.76, SEED < 0.8.
birdFamily_seed("gull",SEED):- SEED >= 0.8, SEED < 0.85.
birdFamily_seed("flycatcher",SEED):- SEED >= 0.85, SEED < 0.90.
birdFamily_seed("swift",SEED):- SEED >= 0.9, SEED < 0.99.
birdFamily_seed("albatross",SEED):- SEED >= 0.99.

color_seed("brown",SEED):- SEED < 0.15.
color_seed("blue",SEED):- SEED >= 0.15, SEED < 0.2.
color_seed("red",SEED):- SEED >= 0.2, SEED < 0.35.
color_seed("golden",SEED):- SEED >= 0.35, SEED < 0.5.
color_seed("crimson",SEED):- SEED >= 0.5, SEED < 0.55.
color_seed("white",SEED):- SEED >= 0.55, SEED < 0.65.
color_seed("black",SEED):- SEED >= 0.65, SEED < 0.75.
color_seed("gray",SEED):- SEED >= 0.75, SEED < 0.8.
color_seed("yellow",SEED):- SEED >= 0.8, SEED < 0.95.
color_seed("violet",SEED):- SEED >= 0.95.

birdPart_seed("tail",SEED):- SEED < 0.15.
birdPart_seed("wing",SEED):- SEED >= 0.15, SEED < 0.2.
birdPart_seed("head",SEED):- SEED >= 0.2, SEED < 0.25.
birdPart_seed("beak",SEED):- SEED >= 0.25, SEED < 0.4.
birdPart_seed("throat",SEED):- SEED >= 0.4, SEED < 0.55.
birdPart_seed("shoulder",SEED):- SEED >= 0.55, SEED < 0.7.
birdPart_seed("breast",SEED):- SEED >= 0.7, SEED < 0.8.
birdPart_seed("crest",SEED):- SEED >= 0.8, SEED < 0.9.
birdPart_seed("neck",SEED):- SEED >= 0.9.

clime_seed("tropical",SEED):- SEED < 0.15.
clime_seed("arctic",SEED):- SEED >= 0.15, SEED < 0.3.
clime_seed("coastal",SEED):- SEED >= 0.3, SEED < 0.55.
clime_seed("prairie",SEED):- SEED >= 0.55, SEED < 0.65.
clime_seed("river",SEED):- SEED >= 0.65, SEED < 0.75.
clime_seed("forest",SEED):- SEED >= 0.75, SEED < 0.85.
clime_seed("mountain",SEED):- SEED >= 0.85.

cardDir_seed("north",SEED):- SEED < 0.25.
cardDir_seed("south",SEED):- SEED >= 0.25, SEED < 0.5.
cardDir_seed("east",SEED):- SEED >= 0.5, SEED < 0.75.
cardDir_seed("west",SEED):- SEED >= 0.75.

rarity_seed("common",SEED):- SEED < 0.65.
rarity_seed("uncommon",SEED):- SEED >= 0.65, SEED < 0.85.
rarity_seed("rare",SEED):- SEED >= 0.85, SEED < 0.96.
rarity_seed("extinct",SEED):- SEED >= 0.97.

descType_seed(coloredPart,SEED):- SEED < 0.45.
descType_seed(clime, SEED):- SEED >= 0.45, SEED < 0.65.
descType_seed(cardDir, SEED):- SEED >= 0.65, SEED < 0.8.
descType_seed(rarity, SEED):- SEED >= 0.8, SEED < 0.91.
descType_seed(wayOfSinging, SEED) :- SEED >= 0.91.

partFeature_seed("speckled",SEED):- SEED < 0.2.
partFeature_seed("mottled",SEED):- SEED >= 0.2, SEED < 0.3.
partFeature_seed("tufted",SEED):- SEED >= 0.3, SEED < 0.4.
partFeature_seed("striped",SEED):- SEED >= 0.4, SEED < 0.6.
partFeature_seed("narrow",SEED):- SEED >= 0.6, SEED < 0.8.
partFeature_seed("oversized",SEED):- SEED >= 0.8.

wayOfSinging_seed("silent",SEED):- SEED < 0.01.
wayOfSinging_seed("squawking",SEED):- SEED >= 0.01, SEED < 0.2.
wayOfSinging_seed("singing",SEED):- SEED >= 0.2, SEED < 0.5.
wayOfSinging_seed("chattering",SEED):- SEED >= 0.5, SEED < 0.65.
wayOfSinging_seed("tweeting",SEED):- SEED >= 0.65, SEED < 0.75.
wayOfSinging_seed("screeching",SEED):- SEED >= 0.75, SEED < 0.85.
wayOfSinging_seed("whistling",SEED):- SEED >= 0.85, SEED < 0.95.
wayOfSinging_seed("groaning",SEED):- SEED >= 0.95.

% You are building the birdhouse, and painting it brown; buying from
% Lowe's open air, green-tarp shaded gardening section a bag of
% birdseeds; filling the little bowl with water; filling the floor with
% the seeds; seeing if something comes (hoping it isn't a squirrel;
% throwing dirt clods at it if it is; standing silently behind the tall
% window if not); teaching a chicken to scream.

a_birdFamily(BIRD_FAMILY):- random(S), birdFamily_seed(BIRD_FAMILY,S).
a_color(COLOR):- random(S), color_seed(COLOR,S).
a_birdPart(BIRD_PART):- random(S), birdPart_seed(BIRD_PART,S).
a_clime(CLIME):- random(S), clime_seed(CLIME,S).
a_cardDir(CARD_DIR):- random(S), cardDir_seed(CARD_DIR,S).
a_rarity(RARITY):- random(S), rarity_seed(RARITY,S).
a_descType(DESC_TYPE):- random(S), descType_seed(DESC_TYPE,S).
a_partFeature(PART_FEATURE):- random(S), partFeature_seed(PART_FEATURE,S).
a_wayOfSinging(WAY_OF_SINGING):- random(S), wayOfSinging_seed(WAY_OF_SINGING,S).

/*
 * THIRTEEN WAYS OF LOOKING AT A BLACKBIRD
 * Wallace Stevens
 *
 * I
 * Among twenty snowy mountains,
 * The only moving thing
 * Was the eye of the blackbird.
 *
 * II
 * I was of three minds,
 * Like a tree
 * In which there are three blackbirds
 *
 * III
 * The blackbird whirled in the autumn winds.
 * It was a small part of the pantomime.
 * ...
 */
```

```
% Walking towards the feathered thing a step at a time, crunching no
% twigs, you place the binoculars over your eyes and rub at the central
% dials until you can distinguish leaf from leaf and feather from
% feather in the crook of the old oak. Standing in a city rock doves
% with green glimmering throats twitter around you; under the ornamental
% bridge, gliding mallards congregate; to the dead hedgehog flesh-headed
% vultures swoop. On imagined branches that flicker, wingless blackbirds
% land


an_attribute_desc(family,[A_FAMILY_FOR_THAT_BIRD_AUNTS_UNCLES_SIBLINGS]):-
        a_birdFamily(A_FAMILY_FOR_THAT_BIRD_AUNTS_UNCLES_SIBLINGS).
an_attribute_desc(coloredPart,[A_PART_OF_THAT_BIRD, A_COLOR_OF_A_PART_OF_THAT_BIRD]):-
        a_birdPart(A_PART_OF_THAT_BIRD), a_color(A_COLOR_OF_A_PART_OF_THAT_BIRD).
an_attribute_desc(clime,[A_PREFERRED_KIND_OF_WIND_AND_VEGETATION]):-
        a_clime(A_PREFERRED_KIND_OF_WIND_AND_VEGETATION).
an_attribute_desc(rarity,[WHAT_MULTITUDES]):-
        a_rarity(WHAT_MULTITUDES).
an_attribute_desc(cardDir,[HOLD_A_COMPASS_UNDER_IT]):-
        a_cardDir(HOLD_A_COMPASS_UNDER_IT).
an_attribute_desc(wayOfSinging,[DESCRIBE_ITS_NOISES]):-
        a_wayOfSinging(DESCRIBE_ITS_NOISES).

% There may be other and contaminating rooms: the metaphor-logic
% mentions odd flecks, leaves, wind-rustles, cloud-pockets. Throat
% feathers all ragged and rippling, if you remember interiors

desc_name(coloredPart,[THE_PART,THE_COLOR],THE_DESC_STR):-
        strs_flatten([THE_COLOR,"-",THE_PART,"ed"],THE_DESC_STR).
desc_name(clime, [THE_CLIME], THE_DESC_STR):-
        THE_DESC_STR = THE_CLIME.
desc_name(cardDir, [THE_CARD_DIR], THE_DESC_STR):-
        strs_flatten([THE_CARD_DIR,"ern"],THE_DESC_STR).
desc_name(rarity, [THE_RARITY], THE_DESC_STR):-
        THE_DESC_STR = THE_RARITY.
desc_name(family, [THE_FAMILY], THE_DESC_STR):-
        THE_DESC_STR = THE_FAMILY.
desc_name(wayOfSinging, [THE_WAY_OF_SINGING], THE_DESC_STR):-
        THE_DESC_STR = THE_WAY_OF_SINGING.

% The Juan Fernandez firecrown in the Juan Fernandez Islands, named
% after the Spanish explorer Juan Fernandez, is a hummingbird only
% spotted on the island Isla de Robinson Crusoe, named after Robinson
% Crusoe, owned by Chile which was named "Chile" by the Spanish
% possibly from the Incan naming of a valley "Chili," a corruption of
% the name of a Picunche chief Tili, or which may be named from a
% Mapuche word "chilli" for "where the land ends," a Quechua word
% "chiri" or "tchilli" for "cold" or "snow," or for the onomatopoeic
% "cheele-cheele" for the warble of a bird known as the "trile"


/*
 * THIRTEEN WAYS OF LOOKING AT A BLACKBIRD
 * Wallace Stevens
 *
 * I
 * Among twenty snowy mountains,
 * The only moving thing
 * Was the eye of the blackbird.
 *
 * II
 * I was of three minds,
 * Like a tree
 * In which there are three blackbirds
 *
 * III
 * The blackbird whirled in the autumn winds.
 * It was a small part of the pantomime.
 *
 * IV
 * A man and a woman are one.
 * A man and a woman and a
 * blackbird
 * Are one.
 *
 * V
 * I do not know which to prefer,
 * The beauty of inflections
 * Or the beauty of innuendoes,
 * The blackbird whistling
 * Or just after.
 * ...
 *
 * (I wanted to change the "man" and "woman" in the 4th one to [PERSON],
 * but i don’t know .. it would have been dishonest i guess (maybe a
 * useful reminder that you can watch so many birds’ wings and still be
 * stuck in human time and things))
 *
 * -- I genuinely do not whether my experience, my life, licenses me
 * -- to or to not write about certain violences (because, of course,
 * -- it matters); the experiences are all neither/and, both/or. I will
 * -- Everything I loved here I hate (full analogy between code and
 * -- language; lyricized innuendo; certain old-boy poets; logicism). I
 * -- tell myself I will not let myself love things without knowing why
 * -- I love things, but I continue to become fascinated by, immersed
 * -- in things, without seriously sustaining a critical thought about
 * -- why I so want to immerse in it, what I am trying to go into and
 * -- away from.
 */

% Now, sitting behind your backyard
% window, with an extra page accidentally printed - you get a pencil,
% and start finding the bird’s shape. Taking out the tin khit
% of caked watercolors, you brush in, moving from light to dark in quick
% washes. The bird will fly away before you’re done, but probably no one
% will notice the imprecisions of pigment and proportion its absence
% leaves.

% -- I do not reject my end-of-teenage syntax. I detect an ear in it.

birdPart(BIRD_PART):-
        birdParts(THE_BPARTS),
        member(BIRD_PART,THE_BPARTS).

that_piece_of_that_bird_in_a_color(THAT_PIECE_OF_THAT_BIRD,[THAT_PIECE_OF_THAT_BIRD,IN_A_COLOR]):
-
        birdPart(THAT_PIECE_OF_THAT_BIRD),
        a_color(IN_A_COLOR).

colors_of_parts_of_a_bird(THE_PIECES_IN_SOME_COLORS):-
        birdParts(THE_PIECES_OF_THAT_BIRD),
        maplist(that_piece_of_that_bird_in_a_color,THE_PIECES_OF_THAT_BIRD,THE_PIECES_IN_SOME_COL
ORS).

% Now, it is well known that your blue and my blue may or may not be the
% same blue, but, beyond cliched thought experiment, after examining
% rods and cones and counting them and wave lenghts and performing some
% similar procedures we also know that most birds see some different
% and ultraviolet shades of blue and off-blue and possess plumage
% reflecting some of these ultraviolet colors. You will not see this
% ultraviolet stripe on a chickadee and you will not have a word for it.

name_for(clime,NAME_IT,_,_,LOCATE_IT,_,FIND_ITS_RELATIONS,_):-
        desc_name(clime,LOCATE_IT,NAME_THE_PLACE),
        desc_name(family,FIND_ITS_RELATIONS,NAME_THE_RELATIONS),
        strs_flatten([NAME_THE_PLACE," ",NAME_THE_RELATIONS],NAME_IT).

name_for(cardDir,NAME_IT,_,_,_,POINT,FIND_ITS_RELATIONS,_):-
        desc_name(cardDir,POINT,NAME_THE_DIR),
        desc_name(family,FIND_ITS_RELATIONS,NAME_THE_RELATIONS),
        strs_flatten([NAME_THE_DIR," ",NAME_THE_RELATIONS],NAME_IT).

name_for(rarity,NAME_IT,_,COUNT_IT,_,_,FIND_ITS_RELATIONS,_):-
        desc_name(rarity,COUNT_IT,NAME_THE_NUMBER),
        desc_name(family,FIND_ITS_RELATIONS,NAME_THE_RELATIONS),
        strs_flatten([NAME_THE_NUMBER," ",NAME_THE_RELATIONS],NAME_IT).

name_for(coloredPart,NAME_IT,COLOR_IT,_,_,_,FIND_ITS_RELATIONS,_):-
        member(THE_COLORED_PART,COLOR_IT),
        desc_name(coloredPart,THE_COLORED_PART,NAME_THE_COLORED_PART),
        desc_name(family,FIND_ITS_RELATIONS,NAME_THE_RELATIONS),
        strs_flatten([NAME_THE_COLORED_PART," ",NAME_THE_RELATIONS],NAME_IT).

name_for(wayOfSinging,NAME_IT,_,_,_,_,FIND_ITS_RELATIONS,LISTEN_TO_IT):-
        desc_name(wayOfSinging,LISTEN_TO_IT,HEAR_IT),
        desc_name(family,FIND_ITS_RELATIONS,NAME_THE_RELATIONS),
        strs_flatten([HEAR_IT," ",NAME_THE_RELATIONS],NAME_IT).

a_name_ofType(clime,NAME_IT,COLOR_IT,COUNT_IT,LOCATE_IT,POINT,FIND_ITS_RELATIONS,LISTEN_TO_IT):-
        name_for(clime,NAME_IT,COLOR_IT,COUNT_IT,LOCATE_IT,POINT,FIND_ITS_RELATIONS,LISTEN_TO_IT)
.

a_name_ofType(cardDir,NAME_IT,COLOR_IT,COUNT_IT,LOCATE_IT,POINT,FIND_ITS_RELATIONS,LISTEN_TO_IT):
-
        name_for(cardDir,NAME_IT,COLOR_IT,COUNT_IT,LOCATE_IT,POINT,FIND_ITS_RELATIONS,LISTEN_TO_I
T).

a_name_ofType(rarity,NAME_IT,COLOR_IT,COUNT_IT,LOCATE_IT,POINT,FIND_ITS_RELATIONS,LISTEN_TO_IT):-
        name_for(rarity,NAME_IT,COLOR_IT,COUNT_IT,LOCATE_IT,POINT,FIND_ITS_RELATIONS,LISTEN_TO_IT
).

a_name_ofType(coloredPart,NAME_IT,COLOR_IT,_,_,_,FIND_ITS_RELATIONS,_):-
        a_birdPart(SOME_PART),
        THE_COLORED_PART = [SOME_PART,_],
        member(THE_COLORED_PART,COLOR_IT),
        desc_name(coloredPart,THE_COLORED_PART,NAME_THE_COLORED_PART),
        desc_name(family,FIND_ITS_RELATIONS,NAME_THE_RELATIONS),
        strs_flatten([NAME_THE_COLORED_PART," ",NAME_THE_RELATIONS],NAME_IT).

a_name_ofType(wayOfSinging,NAME_IT,COLOR_IT,COUNT_IT,LOCATE_IT,POINT,FIND_ITS_RELATIONS,LISTEN_TO
_IT):-
        name_for(wayOfSinging,NAME_IT,COLOR_IT,COUNT_IT,LOCATE_IT,POINT,FIND_ITS_RELATIONS,LISTEN
_TO_IT).

% One from the Spring flock on the nearby tree lands in front of you -
% brown-winged, white-throated, large as a butternut squash. When you
% step towards it, it leaps an equal amount back. When you step away, it
% steps the same distance forward. When your steps approach it twice it
% flaps back up into the crowded tree.

a_bird(NAME,COLORS,COUNT,LOCATE,POINT,FIND_RELATIONS,WEIGH_IT,INDEX_IT,HEAR_IT):-
        colors_of_parts_of_a_bird(COLORS),
        an_attribute_desc(rarity,COUNT),
        an_attribute_desc(clime,LOCATE),
        an_attribute_desc(cardDir,POINT),
        an_attribute_desc(family,FIND_RELATIONS),
        an_attribute_desc(wayOfSinging,HEAR_IT),
        a_descType(A_NAME_TYPE),
        random_between(1,10,WEIGH_IT),
        a_name_ofType(A_NAME_TYPE,NAME,COLORS,COUNT,LOCATE,POINT,FIND_RELATIONS,HEAR_IT),
        INDEX_IT >= 1.

a_bird_ofFamily(NAME,COLORS,COUNT,LOCATE,POINT,ITS_RELATIONS,WEIGH_IT,INDEX_IT,HEAR_IT):-
        colors_of_parts_of_a_bird(COLORS),
        an_attribute_desc(rarity,COUNT),
        an_attribute_desc(clime,LOCATE),
        an_attribute_desc(cardDir,POINT),
        an_attribute_desc(wayOfSinging,HEAR_IT),
        a_descType(A_NAME_TYPE),
        random_between(1,10,WEIGH_IT),
        a_name_ofType(A_NAME_TYPE,NAME,COLORS,COUNT,LOCATE,POINT,ITS_RELATIONS,HEAR_IT),
        INDEX_IT >= 1.

list_bird(ABOUT_A_BIRD,NAME,COLORS,COUNT,LOCATE,POINT,FIND_RELATIONS,WEIGH_IT,INDEX_IT,LISTEN_TO_
IT):-
        ABOUT_A_BIRD = [NAME,COLORS,COUNT,LOCATE,POINT,FIND_RELATIONS,WEIGH_IT,INDEX_IT,LISTEN_TO
_IT].

birdList_birdName(LISTED_BIRD,ITS_NAME):-
        nth0(0,LISTED_BIRD,ITS_NAME).

birdList_birdColors(LISTED_BIRD,ITS_COLORS):-
        nth0(1,LISTED_BIRD,ITS_COLORS).

birdList_birdRarity(LISTED_BIRD,ITS_RARITY):-
        nth0(2,LISTED_BIRD,ITS_RARITY).

birdList_birdClime(LISTED_BIRD,ITS_CLIME):-
        nth0(3,LISTED_BIRD,ITS_CLIME).

birdList_birdDir(LISTED_BIRD,ITS_DIR):-
        nth0(4,LISTED_BIRD,ITS_DIR).

birdList_birdFamily(LISTED_BIRD,ITS_FAMILY):-
        nth0(5,LISTED_BIRD,ITS_FAMILY).

birdList_birdSize(LISTED_BIRD,ITS_SIZE):-
        nth0(6,LISTED_BIRD,ITS_SIZE).

birdList_birdIndex(LISTED_BIRD,ITS_SIZE):-
        nth0(7,LISTED_BIRD,ITS_SIZE).

birdList_birdSinging(LISTED_BIRD,ITS_SINGING):-
        nth0(8,LISTED_BIRD,ITS_SINGING).

% To identify a bird quickly, note the shape of the wings in profile,
% black against sunlight; or count toes; forget the background tree,
% forget the ruffle of feathers, but maybe count a flock; dappled
% sunlight is distracting but removing it, as painters but not
% cameras do, the distinction between speckled and striped is a good
% way to tell thrush from thrush; the difference, between 2 and 5 cm, in
% the length of a white brow stripe can identify Siberian from East
% Asian variants of a swallow; plumage patterns of the juvenile are
% different and require another chart.

a_birdList(ABOUT_A_BIRD):-
        a_bird(NAME,COLORS,COUNT,LOCATE,POINT,FIND_RELATIONS,WEIGH_IT,1,SNG),
        list_bird(ABOUT_A_BIRD,NAME,COLORS,COUNT,LOCATE,POINT,FIND_RELATIONS,WEIGH_IT,1,SNG).

a_birdListNum(ABOUT_A_BIRD,THE_INDEX):-
        a_bird(NAME,COLORS,COUNT,LOCATE,POINT,FIND_RELATIONS,WEIGH_IT,THE_INDEX,LISTEN_TO_IT),
        list_bird(ABOUT_A_BIRD,NAME,COLORS,COUNT,LOCATE,POINT,FIND_RELATIONS,WEIGH_IT,THE_INDEX,L
ISTEN_TO_IT).

a_birdList_ofFamilyNum(ITS_RELATIONS,ABOUT_A_BIRD,THE_INDEX):-
        a_bird_ofFamily(NAME,COLORS,COUNT,LOCATE,POINT,ITS_RELATIONS,WEIGH_IT,THE_INDEX,LISTEN_TO
_IT),
        list_bird(ABOUT_A_BIRD,NAME,COLORS,COUNT,LOCATE,POINT,ITS_RELATIONS,WEIGH_IT,THE_INDEX,LI
STEN_TO_IT).

a_relationListNum(OLD_BIRD,RELATED_BIRD,THE_INDEX):-
        birdList_birdFamily(OLD_BIRD,THE_FAM),
        a_birdList_ofFamilyNum(THE_FAM,RELATED_BIRD,THE_INDEX).

% Consider a bird through a window. It’s a small fluffed mass -
% vibrating, folded up. Lacking better names for it, call it A Little
% Brown Thing.

bird_sociability(ITS_NAME,"solitary"):-
        string_length(ITS_NAME,NAME_LENGTH),
        13 >= NAME_LENGTH.

bird_sociability(ITS_NAME,"semi-solitary"):-
        string_length(ITS_NAME,NAME_LENGTH),
        NAME_LENGTH > 13,
        17 >= NAME_LENGTH.

bird_sociability(ITS_NAME,"flocking"):-
```

```
        string_length(ITS_NAME,NAME_LENGTH),
        NAME_LENGTH >= 17.

sociability_descriptor("solitary",DESCRIPTOR):-
        random_member(DESCRIPTOR,["alone","in solitude","singley"]).

sociability_descriptor("semi-solitary",DESCRIPTOR):-
        random_member(DESCRIPTOR,["in pairs","with a few of their kind","individually or in small
 groups"]).

sociability_descriptor("flocking",DESCRIPTOR):-
        random_member(DESCRIPTOR,["with many of their kind","in flocks","in large congregations"]
).

behaviour_type(BEHAVIOUR_INT,"inter-species"):-
        0 is BEHAVIOUR_INT mod 7.

behaviour_type(BEHAVIOUR_INT,"nesting"):-
        1 is BEHAVIOUR_INT mod 7.

behaviour_type(BEHAVIOUR_INT,"flight"):-
        2 is BEHAVIOUR_INT mod 7.

behaviour_type(BEHAVIOUR_INT,"migratory"):-
        3 is BEHAVIOUR_INT mod 7.

behaviour_type(BEHAVIOUR_INT,"mating"):-
        4 is BEHAVIOUR_INT mod 7.

behaviour_type(BEHAVIOUR_INT,"human"):-
        5 is BEHAVIOUR_INT mod 7.

behaviour_type(BEHAVIOUR_INT,"misc"):-
        6 is BEHAVIOUR_INT mod 7.

nesting_material(SEED_INT,THE_MATERIAL):-
        MOD3 is SEED_INT mod 3,
        nth0(MOD3,["dead twigs and leaves","branches","moss, wool, fabric, plant fluff, or string
"],THE_MATERIAL).

nest_type(SEED_INT,NEST_TYPE):-
        MOD5 is SEED_INT mod 5,
        nth0(MOD5,["flat platforms composed of ","suspended structures woven from ","natural cavi
ties in trees filled out with ", "depressions in the ground covered over by ", "tall and loosely
built nests of "],NEST_TYPE).

of_type_behaviour(BEHAVIOUR_INT,"inter-species",ITS_BEHAVIOUR):-
        0 is BEHAVIOUR_INT mod 3,
        a_birdList(SOME_OTHER_BIRD),
        birdList_birdName(SOME_OTHER_BIRD,OTHER_NAME),
        random_member(MINGLE,["mingle with","walk among","stand around"]),
        strs_flatten(["often ",MINGLE," groups of ",OTHER_NAME,"s"],ITS_BEHAVIOUR).

of_type_behaviour(BEHAVIOUR_INT,"inter-species",ITS_BEHAVIOUR):-
        1 is BEHAVIOUR_INT mod 3,
        a_birdList(SOME_OTHER_BIRD),
        birdList_birdName(SOME_OTHER_BIRD,OTHER_NAME),
        strs_flatten(["are found around the homes of the ",OTHER_NAME],ITS_BEHAVIOUR).

of_type_behaviour(BEHAVIOUR_INT,"inter-species",ITS_BEHAVIOUR):-
        2 is BEHAVIOUR_INT mod 3,
        a_birdList(SOME_OTHER_BIRD),
        birdList_birdName(SOME_OTHER_BIRD,OTHER_NAME),
        strs_flatten(["are a brood parasite, leaving their eggs in nests of the ",OTHER_NAME, " r
ather than raising their own young"],ITS_BEHAVIOUR).

of_type_behaviour(BEHAVIOUR_INT,"nesting",ITS_BEHAVIOUR):-
        nesting_material(BEHAVIOUR_INT,THE_MATERIAL),
        nest_type(BEHAVIOUR_INT,NEST_TYPE),
        strs_flatten(["make their homes in ",NEST_TYPE,THE_MATERIAL],ITS_BEHAVIOUR).

of_type_behaviour(BEHAVIOUR_INT,"flight","have lost the capacity for flight"):-
```

```
        0 is BEHAVIOUR_INT mod 3.

of_type_behaviour(BEHAVIOUR_INT,"flight",ITS_BEHAVIOUR):-
        1 is BEHAVIOUR_INT mod 3,
        MOD5 is BEHAVIOUR_INT mod 5,
        nth0(MOD5,["high","low","rapidly","slowly","zigzaggingly"],FLIGHT_KIND),
        strs_flatten(["fly ",FLIGHT_KIND],ITS_BEHAVIOUR).

of_type_behaviour(BEHAVIOUR_INT,"flight",ITS_BEHAVIOUR):-
        2 is BEHAVIOUR_INT mod 3,
        MOD4 is BEHAVIOUR_INT mod 4,
        MOD5 is BEHAVIOUR_INT mod 5,
        nth0(MOD4,["hot","rising","cold","vorticial"],AIR_TYPE),
        nth0(MOD5,["high","low","rapidly","slowly","zigzaggingly"],FLIGHT_KIND),
        strs_flatten(["seek ",AIR_TYPE, " currents of wind to fly ",FLIGHT_KIND, " on"],ITS_BEHAV
IOUR).

of_type_behaviour(BEHAVIOUR_INT,"migratory",ITS_BEHAVIOUR):-
        MOD3 is BEHAVIOUR_INT mod 3,
        MOD4 is BEHAVIOUR_INT mod 4,
        MOD5 is BEHAVIOUR_INT mod 5,
        MOD6 is BEHAVIOUR_INT mod 6,
        cardDirs(CARD_DIRS),
        nth0(MOD4,CARD_DIRS,WHICH_DIR),
        capitalize_first(WHICH_DIR,MIGRATION_DIR),
        nth0(MOD5,["around a pond, across a field, or to the other side of a mountain to the ",
                 "short distances to better nesting grounds further ",
                 "reasonable distances ",
                 "long distances annualy to the same sites in the ",
                 "without rest accross continents and over oceans "],
             MIGRATION_DIST),
        nth0(MOD6,[", to raise their young",", to mate",
                   ", to withstand the weather",
                   ", to find food",", to avoid predators",", to perish"],
             MIGRATION_REASONS),
        nth0(MOD3,["winter","spring","fall"],MIGRATION_SEASON),
        strs_flatten(["travel ",MIGRATION_DIST,MIGRATION_DIR,
                      " in the ",MIGRATION_SEASON, MIGRATION_REASONS],
                     ITS_BEHAVIOUR).

of_type_behaviour(BEHAVIOUR_INT,"human",ITS_BEHAVIOUR):-
        1450 >= BEHAVIOUR_INT,
        MOD3 is BEHAVIOUR_INT mod 3,
        nth0(MOD3,["cities","towns","suburbs"],HUMAN_PLACE),
        strs_flatten(["thrive in ",HUMAN_PLACE],ITS_BEHAVIOUR).

of_type_behaviour(BEHAVIOUR_INT,"human",ITS_BEHAVIOUR):-
        BEHAVIOUR_INT > 1450,
        1550 >= BEHAVIOUR_INT,
        MOD5 is BEHAVIOUR_INT mod 5,
        nth0(MOD5,["conservation programs","environmental legislation",
                   "the protection of their nesting grounds",
                   "breeding and reintroduction programs",
                   "the returning availability of their preferred foods"],
            HUMAN_HELP),
        strs_flatten(["have begun to rebound due to ",HUMAN_HELP],ITS_BEHAVIOUR).

of_type_behaviour(BEHAVIOUR_INT,"human",ITS_BEHAVIOUR):-
        BEHAVIOUR_INT > 1550,
        1725 >= BEHAVIOUR_INT,
        MOD5 is BEHAVIOUR_INT mod 5,
        nth0(MOD5,["pesticide use","the dissapearance of their prey",
                   "the human settlement of their nesting grounds",
                   "extensive hunting",
                   "unknown but presumably human caused changes"],
            HUMAN_HURT),
        strs_flatten(["have been devastated by ",HUMAN_HURT],ITS_BEHAVIOUR).

of_type_behaviour(BEHAVIOUR_INT,"human",ITS_BEHAVIOUR):-
        BEHAVIOUR_INT > 1725,
        2000 >= BEHAVIOUR_INT,
        MOD3 is BEHAVIOUR_INT mod 3,
        MOD5 is BEHAVIOUR_INT mod 5,
        nth0(MOD3,["farmland","backyards","ranches"],FIRST_OUTSKIRT),
        nth0(MOD5,["lakesides","onto telephone wires","beaches","golf courses",
                   "roofs"],SECOND_OUTSKIRT),
        strs_flatten(["live on human outskirts, sometimes venturing into ",
                      FIRST_OUTSKIRT," or ",SECOND_OUTSKIRT],ITS_BEHAVIOUR).

of_type_behaviour(BEHAVIOUR_INT,"human","only survive far from human populations"):-
        BEHAVIOUR_INT > 2000.

of_type_behaviour(_,"mating","mate at the correct times with great theatrics and zeal").

of_type_behaviour(BEHAVIOUR_INT,"misc","seem to to do nothing at all"):-
        0 is BEHAVIOUR_INT mod 2,
        0 is BEHAVIOUR_INT mod 3.

of_type_behaviour(BEHAVIOUR_INT,"misc","are very shy birds, almost impossible to approach"):-
        1 is BEHAVIOUR_INT mod 2,
        0 is BEHAVIOUR_INT mod 3.

of_type_behaviour(BEHAVIOUR_INT,"misc","will congregate and gratefully eat if fed"):-
        0 is BEHAVIOUR_INT mod 2,
        1 is BEHAVIOUR_INT mod 3.

of_type_behaviour(BEHAVIOUR_INT,"misc","will steal sandwiches"):-
        1 is BEHAVIOUR_INT mod 2,
        1 is BEHAVIOUR_INT mod 3.

of_type_behaviour(BEHAVIOUR_INT,"misc","can always find their way home"):-
        0 is BEHAVIOUR_INT mod 2,
        2 is BEHAVIOUR_INT mod 3.

of_type_behaviour(BEHAVIOUR_INT,"misc","collect bright trinkets found in the dirt"):-
        1 is BEHAVIOUR_INT mod 2,
        2 is BEHAVIOUR_INT mod 3.


of_type_behaviour(_,_,"").

bird_behaviour(ITS_NAME,ITS_BEHAVIOUR):-
        string_codes(ITS_NAME,ITS_NUMBERS),
        sum_list(ITS_NUMBERS,BEHAVIOUR_INT),
        behaviour_type(BEHAVIOUR_INT,BEHAVIOUR_TYPE),
        of_type_behaviour(BEHAVIOUR_INT,BEHAVIOUR_TYPE,ITS_BEHAVIOUR).

/*
 * THIRTEEN WAYS OF LOOKING AT A BLACKBIRD
 * Wallace Stevens
 *
 * I
 * Among twenty snowy mountains,
 * The only moving thing
 * Was the eye of the blackbird.
 *
 * II
 * I was of three minds,
 * Like a tree
 * In which there are three blackbirds
 *
 * III
 * The blackbird whirled in the autumn winds.
 * It was a small part of the pantomime.
 *
 * IV
 * A man and a woman are one.
 * A man and a woman and a
 * blackbird
 * Are one.
 *
 * V
 * I do not know which to prefer,
 * The beauty of inflections
 * Or the beauty of innuendoes,
 * The blackbird whistling
 * Or just after.
 *
 * VI
 * Icicles filled the long window
 * With barbaric glass.
 * The shadow of the blackbird
 * Crossed it, to and fro.
 * The mood
 * Traced in the shadow
 * An indecipherable cause.
 *
 * VII
 * O think men of Haddam,
 * Why do you imagine golden birds?
 * Do you not see how the blackbird
 * Walks around the feet
 * Of the women about you
 *
 * VIII
 * I know noble accents
 * And lucid, inescapable rhythms;
 * But I know, too,
 * That the blackbird is involved
 * In what I know.
 * ...
 *
 * (same conflicts and confusions present themselves in VII; the same
 * easy and unsatisfying answers (new beaks))
 *
 */

% One weighs a pile of feathers against an egg; measures wingspan in
% talons; looks for stripes and counts spots; compares speckling and
% distinguishes blackbirds’ black hues. (Names leak, it is well known.
% That somehow systems of glimpses, observed differences in the
% sharpness of the V or ^ of the bird’s wings, still aggregate and knot
% into holding language)

comparison_intensifier(NUM1,NUM2,INTS_STR):-
        integer(NUM1), integer(NUM2),
        DIFF = abs(NUM1 - NUM2),
        0 = DIFF,
        INTS_STR = " not".

comparison_intensifier(NUM1,NUM2,INTS_STR):-
        integer(NUM1), integer(NUM2),
        DIFF = abs(NUM1 - NUM2),
        0 < DIFF, DIFF =< 2,
        INTS_STR = " slightly".

comparison_intensifier(NUM1,NUM2,INTS_STR):-
        integer(NUM1), integer(NUM2),
        DIFF = abs(NUM1 - NUM2),
        2 < DIFF, DIFF =< 3,
        INTS_STR = " somewhat".

comparison_intensifier(NUM1,NUM2,INTS_STR):-
        integer(NUM1), integer(NUM2),
        DIFF = abs(NUM1 - NUM2),
        3 < DIFF, DIFF =< 4,
        INTS_STR = "".

comparison_intensifier(NUM1,NUM2,INTS_STR):-
        integer(NUM1), integer(NUM2),
        DIFF = abs(NUM1 - NUM2),
        4 < DIFF,
        INTS_STR = " much".

comparison_str(NUM1,NUM2,COMP_STR):-
        integer(NUM1), integer(NUM2),
        NUM1 > NUM2,
        comparison_intensifier(NUM1,NUM2,INTS_STR),
        strs_flatten([INTS_STR," larger than"],COMP_STR).

comparison_str(NUM1,NUM2,COMP_STR):-
        integer(NUM1), integer(NUM2),
        NUM1 < NUM2,
        comparison_intensifier(NUM1,NUM2,INTS_STR),
        strs_flatten([INTS_STR," smaller than"],COMP_STR).

comparison_str(NUM1,NUM2,COMP_STR):-
        integer(NUM1), integer(NUM2),
        NUM1 = NUM2,
        COMP_STR = " as large as".

compare_colors(FIRST_COLORS, SECOND_COLORS, IN_COMMON):-
        intersection(FIRST_COLORS, SECOND_COLORS, IN_COMMON).

% And now, a brief break for ode-singing -

bird_chirps(["ai","ou","ka","rik","chi","er","tee","oo",
             "wee","kraa","coo","kyik","ah","per","pip",
             "chip","klip","kok"]).

% Ode to a Nightingale
%    John Keats
%
% My heart aches, and a drowsy numbness pains
%    My sense, as though of hemlock I had drunk,
% Or emptied some dull opiate to the drains
%    One minute past, and Lethe-wards had sunk:
% 'Tis not through envy of thy happy lot
%    But being too happy in thine happiness,-
%       That though, light-winged Dryad of the trees
%          In some melodious plot
%       Of beechen green, and shadows numberless,
%          Singest of summer in full-throated ease.
%
% O, for a drought of vintage! that hath been
%    Cool'd a long age in the deep-delved earth,
% Tasting of Flora and the country green,
%    Dance, and Provencal song, and sunburnt mirth!
% O for a beaker full of the warm South,
%    Full of the true, the blushful Hippocrene,
%       With beaded bubbles winking at the brim,
%          And purple-stained mouth;
%    That I might drink, and leave the world unseen,
%       And with thee fade away into the forest dim:
%
% Fade far away, dissolve, and quite forget
%    What thou among the leaves hast never known,
% The weariness, the fever, and the fret
%    Here, where men sit and hear each other groan;
% Where palsy shakes a few, sad, last gray hairs,
%    Where youth grows pale, and spectre-thin, and dies;
%       Where but to think is to be full of sorrow
%          And leaden-eyed despairs,
%       Where Beauty cannot keep her lustrous eyes,
%         Or new Love pine at them beyond to-morrow.
%
% Away! away! for I will fly to thee,
%    Not charioted by Bacchus and his pards,
% But on the viewless wings of Poesy,
%    Though the dull brain perlexes and retards:
% Already with thee! tender is the night,
%    And haply the Queen-Moon is on her throne,
%       Cluster'd around by all her starry Fays;
%          But here there is no light,
%    Save what from heaven is with the breezes blown
%       Through verdurous glooms and winding mossy ways
%
% I cannot see what flowers are at my feet,
%    Now what soft incense hangs upon the boughs,
% But, in embalmed darkness, guess each sweet
%    Wherewith the seasonable month endows
% The grass, the thicket, and the fruit-tree wild;
%    White hawthorn and the pastoral eglantine;
%       Fast fading violets cover’d up in leaves;
%          And mid-May’s eldest child,
%    The coming musk-rose, full of dewy wine,
%       The murmurous haunt of flies on summer eves.
%
% Darkling I listen; and, for many a time
%    I have been half in love with easeful Death,
% Call’d him soft names in many a mused rhyme,
%    To take into the air my quiet breath;
%       Now more then ever seems it rich to die,
%    To cease upon the midnight with no pain,
%       While thou art pouring forth thy soul abroad
%          In such an ecstasy!
%    Still wouldst thou sing, and I have ears in vain-
%       To thy high requiem become a sod.
%
% Thou wast not born for death, immortal Bird!
%    No hungry generations thread thee down;
% The voice I hear this passing night was heard
%    In ancient days by emperor and clown:
% Perhaps the self-same song that found a path
%    Through the sad heart of Ruth, when, sick for home,
%       She stood in tears amid the alien corn;
%          The same that oft-times hath at intervels.
%
%
%    Charm’d magic casements, opening on the foam
%       Of perilous seas in faery lands forlorn.
%
% Forlorn! the very word is like a bell
%    To toll me back from thee to my sole self!
% Adieu! the fancy cannot cheat so well
%    As she is fam’d to do, deceiving elf.
% Adieu! adieu! thy plaintive anthem fades
%    Past the near meadows, over the still stream,
%       Up the hill-side; and now ’tis buried deep
%          In the next valley-glades:
%    Was it a vision, or a waking dream?
%       Fled is that music:-Do I wake or sleep?

word --> [chirp].

tweet --> [first_chirp], word.
tweet --> [first_chirp], word, word.

melody --> tweet.
melody --> [shriek], tweet.

structure --> [echo_shriek].
structure --> [but_with_a_chirp].
structure --> [].
structure --> [chirp_echo].

song --> melody, structure.
song --> melody, [pause], song.

% The sound of beak on wood,
% clear air, the sun-hot dirt;
% quick clicks frantic as footsteps.
%
% Elsewhere a chirping,
% as microwaves or cellphones do.

makeSomeSound(THE_SOUND):-
        bird_chirps(ALL_POSSIBLE_SOUNDS),
        random_member(THE_SOUND,ALL_POSSIBLE_SOUNDS).
```

```
parse_song([],SUNG_SONG,SUNG_SONG).
parse_song([first_chirp|REST_OF_SONG],SUNG_SO_FAR,WHOLE_SUNG_SONG):-
        makeSomeSound(THE_SOUND),
        strs_flatten([SUNG_SO_FAR,THE_SOUND],MORE_SUNG),
        parse_song(REST_OF_SONG,MORE_SUNG,WHOLE_SUNG_SONG).
parse_song([chirp|REST_OF_SONG],SUNG_SO_FAR,WHOLE_SUNG_SONG):-
        makeSomeSound(THE_SOUND),
        strs_flatten([SUNG_SO_FAR,"-",THE_SOUND],MORE_SUNG),
        parse_song(REST_OF_SONG,MORE_SUNG,WHOLE_SUNG_SONG).
parse_song([pause|REST_OF_SONG],SUNG_SO_FAR,WHOLE_SUNG_SONG):-
        strs_flatten([SUNG_SO_FAR," "],SUNG_PAUSED),
        parse_song(REST_OF_SONG,SUNG_PAUSED,WHOLE_SUNG_SONG).
parse_song([shriek|REST_OF_SONG],SUNG_SO_FAR,WHOLE_SUNG_SONG):-
        makeSomeSound(THE_SOUND),
        string_upper(THE_SOUND,LOUD_SOUND),
        strs_flatten([SUNG_SO_FAR,LOUD_SOUND],MORE_SUNG),
        parse_song(REST_OF_SONG,MORE_SUNG,WHOLE_SUNG_SONG).

parse_song([echo_shriek],SUNG,WHOLE_SONG):-
        makeSomeSound(THE_SOUND),
        string_upper(THE_SOUND,LOUD_SOUND),
        strs_flatten([SUNG," ",SUNG,LOUD_SOUND],WHOLE_SONG).
parse_song([but_with_a_chirp],SUNG,WHOLE_SONG):-
        makeSomeSound(THE_SOUND),
        strs_flatten([SUNG," ",THE_SOUND],WHOLE_SONG).
parse_song([chirp_echo],SUNG,WHOLE_SONG):-
        makeSomeSound(THE_SOUND),
        strs_flatten([SUNG," ",THE_SOUND," ",SUNG],WHOLE_SONG).

song_sung(SONG,SUNG):-
        parse_song(SONG,"",SUNG).

% Some of the things birds are typically about are freedom or grace, but
% also the cackle of their crowd or the ugly grub in their mouth. One of
% the things i wonder about then, about them, is about the distinction
% between squawk and song, and who makes it and who hears it.
%
% The hope is to stand in hot pollen-laden air as the sharp sections
% of pitch thrown about follow and reiterate each other into music or
% commotion.
%
% (So sing thing, sing)
% (Or squawk flock squawk, i suppose, at your
% discretion and preference)

nth_song(0,[]).

nth_song(N,NTH_SONG):-
        findnsols(N,THIS_SONG,song(THIS_SONG,[]),THE_SONGS),
        NMONE is N - 1,
        nth0(NMONE,THE_SONGS,NTH_CHIRPS),
        song_sung(NTH_CHIRPS,NTH_SONG).

bird_birdSong(LISTED_BIRD,THE_SONG):-
        birdList_birdIndex(LISTED_BIRD,THE_NUMBER),
        nth_song(THE_NUMBER,THE_SONG).

% When they stop writing about love and death, they find that they are
% walking in a forest; I can’t tell you who they are or hear what they
% are saying, but I can tell you about the leaves on the branches and
% the loam and orange light. Around them there are: 2 woodpeckers. 1
% swallow. A vulture overhead and a swift in a bush.

part_phrase(PART,LIST_THE_BIRD,TALK_ABOUT_THAT):-
        birdList_birdColors(LIST_THE_BIRD,ALL_THE_BIRD),
        member([PART,ITS_COLOR], ALL_THE_BIRD),
        strs_flatten([ITS_COLOR," ",PART],TALK_ABOUT_THAT).

describe_that_part([PART,ITS_COLOR],THAT_PART_DESCRIBED):-
        strs_flatten([" a ",ITS_COLOR, " ", PART],THAT_PART_DESCRIBED).

describe_parts([],"").
describe_parts([A_PART],DESCRIBE_IT):-
        describe_that_part(A_PART,DESCRIBE_IT).

describe_parts([A_PART | OTHER_PARTS],DESCRIBE_THEM):-
        length(OTHER_PARTS,NUM_OTHER_PARTS),
        NUM_OTHER_PARTS > 0,
        describe_that_part(A_PART,DESCRIBE_FIRST),
        describe_parts(OTHER_PARTS,DESCRIBE_REST),
        strs_flatten([DESCRIBE_FIRST," and", DESCRIBE_REST], DESCRIBE_THEM).

% In the Splash Zone of the Monterey Bay Aquarium, the only exhibit to
% feature international fish (as children require their foreign color
% and striping), on the way to their room of penguins given both water
% and a rocky plaster beach sans ice floe, there is a maze of coral and
% anemonoes, where you, the child, may redirect the flow and crash of
% the waves by placing blocks; all this all is is a bunch of switches,
% but the question, i think, is, how we can we make, or make a better,
% blackbird with them? How can we make the bird exuberant or melancholy?
% How can we direct the next wave to splash our cousin’s face?

numIdSentTypes(6).

typically_synonym(TYPICALLY_SYNONYM):-
        random_member(TYPICALLY_SYNONYM,["typically","usually","generally",
                                         "most often"]).
identified_synonym(IDENTIFIED_SYNONYM):-
        random_member(IDENTIFIED_SYNONYM,["identified","distinguished","told apart"]).
notable_synonym(NOTABLE_SYNONYM):-
        random_member(NOTABLE_SYNONYM,["notable","conspicuous","remarkable"]).

identificatoryInfo_sentence(PART1_FEATURE, PART1, PART2_FEATURE, PART2, ID_SENT):-
        numIdSentTypes(NUM_SENT_TYPES),
        random_between(1,NUM_SENT_TYPES,SENT_TYPE),
        identificatoryInfo_sentence(SENT_TYPE,PART1_FEATURE, PART1, PART2_FEATURE,
                                    PART2, ID_SENT).

identificatoryInfo_sentence(1,PART1_FEATURE, PART1, PART2_FEATURE, PART2, ID_SENT):-
        identified_synonym(IDENTIFIED),
        strs_flatten(["One can be ",IDENTIFIED," by its ", PART1_FEATURE," ", PART1,
                      " and its’ ", PART2_FEATURE," ",PART2,". "],ID_SENT).

identificatoryInfo_sentence(2,PART1_FEATURE, PART1, _, _, ID_SENT):-
        notable_synonym(NOTABLE),
        strs_flatten(["Their ", PART1_FEATURE," ", PART1, "s are ",NOTABLE,". "],ID_SENT).

identificatoryInfo_sentence(3,PART1_FEATURE, PART1, PART2_FEATURE, PART2, ID_SENT):-
        strs_flatten(["They have ", PART1_FEATURE," ", PART1, "s and ",
                      PART2_FEATURE," ",PART2,"s. "],ID_SENT).

identificatoryInfo_sentence(4,PART1_FEATURE, PART1, _, PART2, ID_SENT):-
        typically_synonym(TYPICALLY),
        identified_synonym(IDENTIFIED),
        strs_flatten(["They can ",TYPICALLY," be ",IDENTIFIED," by their ",
                      PART1_FEATURE," ", PART1,"s and ",PART2,"s. "],ID_SENT).

identificatoryInfo_sentence(5,PART1_FEATURE, PART1, _, _, ID_SENT):-
        typically_synonym(TYPICALLY),
        notable_synonym(NOTABLE),
        strs_flatten(["Its’ ",NOTABLE," ",PART1_FEATURE," ", PART1, " ",
                      TYPICALLY," lets you identify one. "],ID_SENT).

identificatoryInfo_sentence(6,PART1_FEATURE, PART1, PART2_FEATURE, PART2, ID_SENT):-
        typically_synonym(TYPICALLY_LOWER),
        capitalize_first(TYPICALLY_LOWER,TYPICALLY),
        notable_synonym(NOTABLE),
        identified_synonym(IDENTIFIED),
        strs_flatten([TYPICALLY,", one can be ",IDENTIFIED," by its ",PART1_FEATURE,
                      " ", PART1, " and its’ ",NOTABLE," ",PART2_FEATURE," ",PART2,". "],
                     ID_SENT).

identificatory_sentence(_,ID_SENT):-
```

```
        birdParts(B_PARTS),
        random_member(PART1,B_PARTS),
        delete(B_PARTS,PART1,OTHER_B_PARTS),
        random_member(PART2,OTHER_B_PARTS),
        a_partFeature(PART1_FEATURE),
        a_partFeature(PART2_FEATURE),
        identificatoryInfo_sentence(PART1_FEATURE, PART1, PART2_FEATURE, PART2, ID_SENT).

% There are estimated to be 19,000,000 mallards to 50 crested
% shellducks. Between 350 and 1500 scarlet banded barbets on the summit
% of a solitary Peruvian mountain to between 73.5 million and 216
% million great spotted woodpeckers, over 40,000,000 willow ptarmigans
% and 1679 flightless cormorants. Some billions, maybe hundreds of
% billions, overall. Someone sits on the rocky beach in rain-pants
% twisting bands round wet-feathered legs.

rarity_advRarity("common","often").
rarity_advRarity("uncommon","sometimes").
rarity_advRarity("rare","occasionally").
rarity_advRarity("extinct","never").


clime_somePlace(THE_CLIME,THE_PLACE):-
        clime_climePlaces(THE_CLIME,POSSIBLE_PLACES),
        random_member(THE_PLACE,POSSIBLE_PLACES).

clime_someOtherPlace(THE_CLIME,THAT_PLACE,ANOTHER_PLACE):-
        clime_climePlaces(THE_CLIME,POSSIBLE_PLACES),
        delete(POSSIBLE_PLACES,THAT_PLACE,REMAINING_PLACES),
        random_member(ANOTHER_PLACE,REMAINING_PLACES).

distributionAndDiet_sentence(LISTED_BIRD,DIST_SENT):-
        birdList_birdName(LISTED_BIRD,NAME_FOR_EM),
        birdList_birdRarity(LISTED_BIRD,[HOW_MANY]),
        birdList_birdDir(LISTED_BIRD,POINT_WHERE),
        birdList_birdClime(LISTED_BIRD,[WHAT_WEATHER]),
        birdList_birdFamily(LISTED_BIRD,[ITS_RELATIONS]),
        birdFamily_diet(ITS_RELATIONS,DIET),
        clime_somePlace(WHAT_WEATHER,WHERE),
        clime_someOtherPlace(WHAT_WEATHER,WHERE,WHERE_ELSE),
        diet_habitat_foodSource(DIET,WHERE,FOOD),
        rarity_advRarity(HOW_MANY,HOW_MANYLY),
        desc_name(cardDir,POINT_WHERE,WHERE_POINTED),
        distributionDietInfo_sentence(NAME_FOR_EM,HOW_MANYLY,WHERE_POINTED,WHERE,
                                      WHERE_ELSE,FOOD,DIST_SENT_ALMOST),
        capitalize_first(DIST_SENT_ALMOST,DIST_SENT).

numDistSentTypes(7).

residein_synonym(RESIDEIN_SYNONYM):-
        random_member(RESIDEIN_SYNONYM,["reside in","live in","occupy","inhabit"]).
spotted_synonym(SPOTTED_SYNONYM):-
        random_member(SPOTTED_SYNONYM,["spotted","found","seen","observed","found"]).
feedingon_synonym(FEEDINGON_SYNONYM):-
        random_member(FEEDINGON_SYNONYM,["feeding on","eating","consuming"]).

distributionDietInfo_sentence(THEM_NAMED,HOW_MANYLY,WHERE_POINTED,WHERE,WHERE_ELSE,
                              FOOD,DIST_SENT):-
        numDistSentTypes(NUM_SENT_TYPES),
        random_between(1,NUM_SENT_TYPES,SENT_TYPE),
        distributionDietInfo_sentence(SENT_TYPE,THEM_NAMED,HOW_MANYLY,WHERE_POINTED,
                                      WHERE,WHERE_ELSE,FOOD,DIST_SENT).

distributionDietInfo_sentence(1,THEM_NAMED,HOW_MANYLY,WHERE_POINTED,WHERE,WHERE_ELSE,
                              FOOD,DIST_SENT):-
        typically_synonym(TYPICALLY),
        residein_synonym(RESIDEIN),
        strs_flatten([THEM_NAMED,"s ",HOW_MANYLY," ",RESIDEIN," ",WHERE_POINTED," ",
                      WHERE_ELSE," or ",WHERE," where they ",TYPICALLY,
                      " subsist on ",FOOD,". "],DIST_SENT).

distributionDietInfo_sentence(2,THEM_NAMED,HOW_MANYLY,WHERE_POINTED,WHERE,
                              WHERE_ELSE,_,DIST_SENT):-
        spotted_synonym(SPOTTED),
        strs_flatten([THEM_NAMED,"s can ",HOW_MANYLY," be ",SPOTTED," around ",
                      WHERE_POINTED," ",WHERE," or at times in ",WHERE_POINTED,
                      " ",WHERE_ELSE,". "],DIST_SENT).

distributionDietInfo_sentence(3,THEM_NAMED,HOW_MANYLY,WHERE_POINTED,WHERE,_,FOOD,
                              DIST_SENT):-
        spotted_synonym(SPOTTED),
        feedingon_synonym(FEEDINGON),
        strs_flatten(["You are most likely to discover ",THEM_NAMED,"s in ",
                      WHERE_POINTED," ",WHERE,". There, they can ",HOW_MANYLY," be ",
                      SPOTTED," ",FEEDINGON," ",FOOD,". "],DIST_SENT).

distributionDietInfo_sentence(4,THEM_NAMED,HOW_MANYLY,WHERE_POINTED,WHERE,_,FOOD,DIST_SENT):-
        spotted_synonym(SPOTTED),
        strs_flatten([THEM_NAMED,"s may ",HOW_MANYLY," be ",SPOTTED," in ",WHERE_POINTED," ",
                      WHERE," searching for ",FOOD,". "],DIST_SENT).

distributionDietInfo_sentence(5,THEM_NAMED,_,WHERE_POINTED,WHERE,_,FOOD,DIST_SENT):-
        strs_flatten([THEM_NAMED,"s feed on ",FOOD," in ",WHERE_POINTED," ",WHERE,". "],
                     DIST_SENT).

distributionDietInfo_sentence(6,THEM_NAMED,HOW_MANYLY,WHERE_POINTED,WHERE,WHERE_ELSE,
                              FOOD,DIST_SENT):-
        spotted_synonym(SPOTTED),
        strs_flatten([THEM_NAMED,"s may ",HOW_MANYLY," be ",SPOTTED," in ",WHERE_POINTED, " ",
                      WHERE_ELSE," or in ",WHERE,". Their diet consists primarily of ",FOOD,". "]
,
                     DIST_SENT).

distributionDietInfo_sentence(7,THEM_NAMED,HOW_MANYLY,WHERE_POINTED,WHERE,WHERE_ELSE,
                              FOOD,DIST_SENT):-
        spotted_synonym(SPOTTED),
        strs_flatten([THEM_NAMED,"s can ",HOW_MANYLY," be ",SPOTTED," eating ",FOOD," in ",
                      WHERE_POINTED," ",WHERE,", or now and then in ",WHERE_ELSE,". "],
                     DIST_SENT).

comparative_phrase(LIST_BIRD1,LIST_BIRD2,COMP_PHR):-
        birdList_birdSize(LIST_BIRD1,B1_SIZE),
        birdList_birdSize(LIST_BIRD2,B2_SIZE),
        comparison_str(B1_SIZE,B2_SIZE,COMP_STR),
        strs_flatten([" is", COMP_STR],COMP_PHR).

numCompSentTypes(6).

comparativeInfo_sentence(B1_NAME,B2_NAME,COMP_PHR,DESC1,DESC2,COMP_SENT):-
        numCompSentTypes(NUM_SENT_TYPES),
        random_between(1,NUM_SENT_TYPES,SENT_TYPE),
        comparativeInfo_sentence(SENT_TYPE,B1_NAME,B2_NAME,COMP_PHR,DESC1,DESC2,COMP_SENT).

with_synonym(WITH_SYNONYM):-
        random_member(WITH_SYNONYM,["with","and has","possessing"]).
mistakenfor_synonym(MISTAKENFOR_SYNONYM):-
        random_member(MISTAKENFOR_SYNONYM,["mistaken for","confused with",
                                           "incorrectly identified as"]).

comparativeInfo_sentence(1,B1_NAME,B2_NAME,COMP_PHR,DESC1,DESC2,COMP_SENT):-
        with_synonym(WITH),
        strs_flatten(["The ",B1_NAME,COMP_PHR," the ",B2_NAME,",  ",WITH," a ",
                      DESC1," and ",DESC2,". "],COMP_SENT).
comparativeInfo_sentence(2,B1_NAME,B2_NAME,COMP_PHR,DESC1,DESC2,COMP_SENT):-
        with_synonym(WITH),
        strs_flatten(["The ",B1_NAME, " resembles the ",B2_NAME," but",COMP_PHR,
                      " it, ",WITH," a ",DESC1," and ",DESC2,". "],COMP_SENT).
comparativeInfo_sentence(3,B1_NAME,B2_NAME,COMP_PHR,DESC1,_,COMP_SENT):-
        strs_flatten(["With a ",DESC1,", the ",B1_NAME,COMP_PHR,
                      " the ",B2_NAME,". "],COMP_SENT).

comparativeInfo_sentence(4,B1_NAME,B2_NAME,COMP_PHR,DESC1,DESC2,COMP_SENT):-
        strs_flatten(["The ",B1_NAME,COMP_PHR," the ",B2_NAME," and is notable for its ",
                      DESC1," and ",DESC2,". "],COMP_SENT).
```

```
comparativeInfo_sentence(5,B1_NAME,B2_NAME,COMP_PHR,DESC1,DESC2,COMP_SENT):-
        with_synonym(WITH),
        strs_flatten(["A relative of the ", B2_NAME, ", the ", B1_NAME, COMP_PHR,
                      " it, ",WITH," a ",DESC1," and ",DESC2,". "],COMP_SENT).
comparativeInfo_sentence(6,B1_NAME,B2_NAME,COMP_PHR,DESC1,DESC2,COMP_SENT):-
        mistakenfor_synonym(MISTAKENFOR),
        strs_flatten(["The ", B1_NAME," is sometimes ",MISTAKENFOR," the ", B2_NAME,
                      " and", COMP_PHR, " it. The  ",B1_NAME," has a ",DESC1," and ",DESC2,". "],
COMP_SENT).

comparative_sentence(LIST_BIRD1,LIST_BIRD2,COMP_SENT):-
        birdList_birdName(LIST_BIRD1,B1_NAME),
        birdList_birdName(LIST_BIRD2,B2_NAME),
        birdParts(B_PARTS),
        random_member(PART1,B_PARTS),
        delete(B_PARTS,PART1,OTHER_B_PARTS),
        random_member(PART2,OTHER_B_PARTS),
        part_phrase(PART1,LIST_BIRD1,DESC1),
        part_phrase(PART2,LIST_BIRD1,DESC2),
        comparative_phrase(LIST_BIRD1,LIST_BIRD2,COMP_PHR),
        comparativeInfo_sentence(B1_NAME,B2_NAME,COMP_PHR,DESC1,DESC2,COMP_SENT).

numDescSentTypes(6).

descriptiveInfo_sentence(BIRD_NAME,BFAMILY,DESC1,DESC2,DESC_SENT):-
        numDescSentTypes(NUM_SENT_TYPES),
        random_between(1,NUM_SENT_TYPES,SENT_TYPE),
        descriptiveInfo_sentence(SENT_TYPE,BIRD_NAME,BFAMILY,DESC1,DESC2,DESC_SENT).

descriptiveInfo_sentence(1,BIRD_NAME,_,DESC1,DESC2,DESC_SENT):-
        strs_flatten(["The ",BIRD_NAME," has a ",DESC1," and a ",DESC2,". "],DESC_SENT).

descriptiveInfo_sentence(2,BIRD_NAME,_,DESC1,DESC2,DESC_SENT):-
        notable_synonym(NOTABLE),
        strs_flatten(["The ",BIRD_NAME," is ",NOTABLE," for its ",DESC1," and its ",
                      DESC2,". "],DESC_SENT).

descriptiveInfo_sentence(3,BIRD_NAME,[ITS_FAMILY],DESC1,_,DESC_SENT):-
        strs_flatten(["A ",ITS_FAMILY," with a ",DESC1," is the ",BIRD_NAME,". "],
                     DESC_SENT).

descriptiveInfo_sentence(4,BIRD_NAME,[ITS_FAMILY],DESC1,DESC2,DESC_SENT):-
        strs_flatten(["The ",BIRD_NAME," is a ",ITS_FAMILY, " with a ", DESC1,
                      " and a ", DESC2,". "],DESC_SENT).

descriptiveInfo_sentence(5,BIRD_NAME,[ITS_FAMILY],DESC1,DESC2,DESC_SENT):-
        notable_synonym(NOTABLE),
        strs_flatten(["One kind of ",ITS_FAMILY," is the ",BIRD_NAME, ", a bird ",
                      NOTABLE," for its ", DESC1, " and ", DESC2,". "],DESC_SENT).

descriptiveInfo_sentence(6,BIRD_NAME,_,DESC1,DESC2,DESC_SENT):-
        strs_flatten(["A bird with a ", DESC1, " and a ", DESC2," is the ",
                      BIRD_NAME,". "],DESC_SENT).

descriptive_sentence(LISTED_BIRD,DESC_SENT):-
        birdList_birdName(LISTED_BIRD,BIRD_NAME),
        birdParts(B_PARTS),
        birdList_birdFamily(LISTED_BIRD,BFAMILY),
        random_member(PART1,B_PARTS),
        delete(B_PARTS,PART1,OTHER_B_PARTS),
        random_member(PART2,OTHER_B_PARTS),
        part_phrase(PART1,LISTED_BIRD,DESC1),
        part_phrase(PART2,LISTED_BIRD,DESC2),
        descriptiveInfo_sentence(BIRD_NAME,BFAMILY,DESC1,DESC2,DESC_SENT).

numBehSentTypes(6).

behaviour_sentence(LISTED_BIRD,BEH_SENT):-
        birdList_birdName(LISTED_BIRD,BIRD_NAME),
        bird_behaviour(BIRD_NAME,BEHAVIOUR),
        bird_sociability(BIRD_NAME,EXTRAVERSION),
        behaviourInfo_sentence(EXTRAVERSION,BEHAVIOUR,BEH_SENT).

behaviourInfo_sentence(EXTRAVERSION,BEHAVIOUR,BEH_SENT):-
        numBehSentTypes(NUM_SENT_TYPES),
        random_between(1,NUM_SENT_TYPES,SENT_TYPE),
        behaviourInfo_sentence(SENT_TYPE,EXTRAVERSION,BEHAVIOUR,BEH_SENT).

behaviourInfo_sentence(1,EXTRAVERSION,BEHAVIOUR,BEH_SENT):-
        sociability_descriptor(EXTRAVERSION,SOC_DESC),
        capitalize_first(SOC_DESC,CAP_SOC_DESC),
        strs_flatten([CAP_SOC_DESC,", they ",BEHAVIOUR,". "],BEH_SENT).

behaviourInfo_sentence(2,EXTRAVERSION,BEHAVIOUR,BEH_SENT):-
        sociability_descriptor(EXTRAVERSION,SOC_DESC),
        strs_flatten(["A bird living ",SOC_DESC,", they ",BEHAVIOUR,". "],BEH_SENT).

behaviourInfo_sentence(3,EXTRAVERSION,BEHAVIOUR,BEH_SENT):-
        sociability_descriptor(EXTRAVERSION,SOC_DESC),
        strs_flatten(["These birds ",BEHAVIOUR,", ",SOC_DESC,". "],BEH_SENT).


behaviourInfo_sentence(4,EXTRAVERSION,BEHAVIOUR,BEH_SENT):-
        strs_flatten(["These ",EXTRAVERSION," birds ",BEHAVIOUR,". "],BEH_SENT).

behaviourInfo_sentence(5,_,BEHAVIOUR,BEH_SENT):-
        strs_flatten(["They ",BEHAVIOUR,". "],BEH_SENT).

behaviourInfo_sentence(6,EXTRAVERSION,_,BEH_SENT):-
        sociability_descriptor(EXTRAVERSION,SOC_DESC),
        strs_flatten(["These birds live ",SOC_DESC,". "],BEH_SENT).

numComCSentTypes(1).

commonColor_sentence(FIRST_BIRD,SECOND_BIRD,BEH_SENT):-
        birdList_birdColors(FIRST_BIRD,FIRST_COLORS),
        birdList_birdColors(SECOND_BIRD,SECOND_COLORS),
        compare_colors(FIRST_COLORS,SECOND_COLORS,COMMON_PARTS),
        describe_parts(COMMON_PARTS,IN_COMMON),
        commonColorInfo_sentence(IN_COMMON,BEH_SENT).

commonColorInfo_sentence(IN_COMMON,BEH_SENT):-
        numComCSentTypes(NUM_SENT_TYPES),
        random_between(1,NUM_SENT_TYPES,SENT_TYPE),
        commonColorInfo_sentence(SENT_TYPE,IN_COMMON,BEH_SENT).

commonColorInfo_sentence(1,"",COMMONC_SENT):-
        strs_flatten(["Their colors are completely different. "],COMMONC_SENT).

commonColorInfo_sentence(1,IN_COMMON,COMMONC_SENT):-
        string_length(IN_COMMON,AMOUNT_IN_COMMON),
        AMOUNT_IN_COMMON > 0,
        strs_flatten(["Both birds have ",IN_COMMON,". "],COMMONC_SENT).

% 8 COUNT
% Charles Bukowski
%
% from my bed
% I watch
% 3 birds
% on a telephone
% wire.
%
% one flies
% off.
% then
% another.
%
% one is left,
% then
% it too
% is gone
%
```

```
% my typewriter is
% tombstone still.
%
% and I am
% reduced to bird
% watching.
%
% just thought I’d
% let you
% know,
% fucker.

a_style(SOME_STYLE):- random_member(SOME_STYLE,["low","high","tuneful","abrasive",
                                                "melodic","hoarse"]).

song_sentence(LISTED_BIRD,SONG_SENT):-
        bird_birdSong(LISTED_BIRD,THE_SONG),
        birdList_birdSinging(LISTED_BIRD,[THE_WAY_OF_SINGING]),
        a_style(FIRST_STYLE),
        a_style(SECOND_STYLE),
        songInfo_sentence(THE_WAY_OF_SINGING,FIRST_STYLE,SECOND_STYLE,THE_SONG,SONG_SENT).

numSongSentTypes(6).

sortof_synonym(SORT_OF_SYNONYM):- random_member(SORT_OF_SYNONYM,
                                                ["sort of","kind of","type of"]).
soundslike_synonym(SOUNDS_LIKE_SYNONYM):- random_member(SOUNDS_LIKE_SYNONYM,
                                                        ["sounds like","resembles",
                                                         "might be transcribed"]).

songInfo_sentence(THE_WAY_OF_SINGING,FIRST_STYLE,SECOND_STYLE,THE_SONG,SONG_SENT):-
        numSongSentTypes(NUM_SENT_TYPES),
        random_between(1,NUM_SENT_TYPES,SENT_TYPE),
        songInfo_sentence(SENT_TYPE,THE_WAY_OF_SINGING,FIRST_STYLE,SECOND_STYLE,
                          THE_SONG,SONG_SENT).

songInfo_sentence(1,_,_,_,THE_SONG,SONG_SENT):-
        sortof_synonym(SORTOF),
        strs_flatten(["CALL: a ",SORTOF," \"",THE_SONG,"\". "],SONG_SENT).

songInfo_sentence(2,_,FIRST_STYLE,_,THE_SONG,SONG_SENT):-
        strs_flatten(["CALL: a ",FIRST_STYLE," \"",THE_SONG,"\". "],SONG_SENT).

songInfo_sentence(3,WAY_OF_SINGING,FIRST_STYLE,_,THE_SONG,SONG_SENT):-
        soundslike_synonym(SOUNDSLIKE),
        strs_flatten(["CALL: a ",FIRST_STYLE," ",WAY_OF_SINGING," which ",SOUNDSLIKE,
                      " \"",THE_SONG,"\". "],SONG_SENT).

songInfo_sentence(4,WAY_OF_SINGING,FIRST_STYLE,SECOND_STYLE,THE_SONG,SONG_SENT):-
        soundslike_synonym(SOUNDSLIKE),
        strs_flatten(["CALL: a ",WAY_OF_SINGING," that starts ",FIRST_STYLE," and ends ",
                      SECOND_STYLE,", which ",SOUNDSLIKE," \"",THE_SONG,"\". "],SONG_SENT).

songInfo_sentence(5,WAY_OF_SINGING,FIRST_STYLE,SECOND_STYLE,THE_SONG,SONG_SENT):-
        sortof_synonym(SORTOF),
        strs_flatten(["CALL: a ",SORTOF," ",WAY_OF_SINGING," - first ",FIRST_STYLE," then ",
                      SECOND_STYLE," \"",THE_SONG,"\". "],SONG_SENT).

songInfo_sentence(6,_,_,_,THE_SONG,SONG_SENT):-
        strs_flatten(["CALL: a \"",THE_SONG,"\". "],SONG_SENT).


/*
 * THIRTEEN WAYS OF LOOKING AT A BLACKBIRD
 * Wallace Stevens
 *
 * I
 * Among twenty snowy mountains,
 * The only moving thing
 * Was the eye of the blackbird.
 *
 * II
 * I was of three minds,
 * Like a tree
 * In which there are three blackbirds
 *
 * III
 * The blackbird whirled in the autumn winds.
 * It was a small part of the pantomime.
 *
 * IV
 * A man and a woman
 * Are one.
 * A man and a woman and a blackbird
 * Are one.
 *
 * V
 * I do not know which to prefer,
 * The beauty of inflections
 * Or the beauty of innuendoes,
 * The blackbird whistling
 * Or just after.
 *
 * VI
 * Icicles filled the long window
 * With barbaric glass.
 * The shadow of the blackbird
 * Crossed it, to and fro.
 * The mood
 * Traced in the shadow
 * An indecipherable cause.
 *
 * VII
 * O think men of Haddam,
 * Why do you imagine golden birds?
 * Do you not see how the blackbird
 * Walks around the feet
 * Of the women about you
 *
 * VIII
 * I know noble accents
 * And lucid, inescapable rhythms;
 * But I know, too,
 * That the blackbird is involved
 * In what I know.
 *
 * IX
 * When the blackbird flew out of sight,
 * It marked the edge
 * Of one of many circles.
 *
 * X
 * At the sight of blackbirds
 * Flying in green light,
 * Even the bawds of euphony
 * Would cry out sharply.
 *
 * XI
 * He rode over Connecticut
 * In a glass coach.
 * Once, a fear pierced him,
 * In that he mistook
 * The shadow of his equipage
 * For blackbirds.
 *
 * XII
 * The river is moving.
 * The blackbird must be flying.
 *
 * XIII
 * It was evening all afternoon.
 * It was snowing
 * And it was going to snow.
 * The blackbird sat
```

```
 * In the cedar-limbs.
 */

% Some birds names are people’s names. But to invent the names of birds
% named after people would require inventing people, and that is beyond
% the scope of this particular project.

ofBirds_ofTypes_text(_,_,[],"").
ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,[comparative|OTHER_TYPES],THE_TEXT):-
        comparative_sentence(LIST_BIRD1,LIST_BIRD2,COMP_SENT),
        ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,OTHER_TYPES,REMAINING_TEXT),
        string_concat(COMP_SENT,REMAINING_TEXT,THE_TEXT).
ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,[common_colors|OTHER_TYPES],THE_TEXT):-
        commonColor_sentence(LIST_BIRD1,LIST_BIRD2,COMC_SENT),
        ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,OTHER_TYPES,REMAINING_TEXT),
        string_concat(COMC_SENT,REMAINING_TEXT,THE_TEXT).
ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,[identificatory|OTHER_TYPES],THE_TEXT):-
        identificatory_sentence(LIST_BIRD1,ID_SENT),
        ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,OTHER_TYPES,REMAINING_TEXT),
        string_concat(ID_SENT,REMAINING_TEXT,THE_TEXT).
ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,[distribution|OTHER_TYPES],THE_TEXT):-
        distributionAndDiet_sentence(LIST_BIRD1,DIST_SENT),
        ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,OTHER_TYPES,REMAINING_TEXT),
        string_concat(DIST_SENT,REMAINING_TEXT,THE_TEXT).
ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,[descriptive|OTHER_TYPES],THE_TEXT):-
        descriptive_sentence(LIST_BIRD1,DESC_SENT),
        ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,OTHER_TYPES,REMAINING_TEXT),
        string_concat(DESC_SENT,REMAINING_TEXT,THE_TEXT).
ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,[behaviour|OTHER_TYPES],THE_TEXT):-
        behaviour_sentence(LIST_BIRD1,BEH_SENT),
        ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,OTHER_TYPES,REMAINING_TEXT),
        string_concat(BEH_SENT,REMAINING_TEXT,THE_TEXT).
ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,[song|OTHER_TYPES],THE_TEXT):-
        song_sentence(LIST_BIRD1,SONG_SENT),
        ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,OTHER_TYPES,REMAINING_TEXT),
        string_concat(SONG_SENT,REMAINING_TEXT,THE_TEXT).
ofBirds_ofTypes_text(LIST_BIRD,_,[nothing|OTHER_TYPES],THE_TEXT):-
        ofBird_ofTypes_text(LIST_BIRD,OTHER_TYPES,REMAINING_TEXT),
        birdList_birdName(LIST_BIRD, HAS_A_NAME),
        strs_flatten(["We know nothing about the ", HAS_A_NAME,"."], NOTHING_SENT),
        string_concat(NOTHING_SENT,REMAINING_TEXT,THE_TEXT).



% For a birdwatcher, we ought to say something about the males and
% females - more often than not one large, one small, one resplendant,
% one gray fluff - but for nonexistent birds we will do nothing of the
% sort.

compTextPatterns([[comparative,identificatory,distribution,behaviour,song],
                  [comparative,common_colors,distribution,behaviour,song],
                  [comparative,distribution,behaviour,song],
                  [comparative,distribution,identificatory,behaviour,song],
                  [nothing]]).

comparative_text([LIST_BIRD1,LIST_BIRD2],THE_TEXT):-
        birdList_birdName(LIST_BIRD1,A_NAME),
        string_upper(A_NAME,A_TITLE),
        compTextPatterns(COMP_TEXT_PATTERNS),
        random_member(COMP_TEXT_PATTERN,COMP_TEXT_PATTERNS),
        ofBirds_ofTypes_text(LIST_BIRD1,LIST_BIRD2,COMP_TEXT_PATTERN,COMP_TEXT),
        strs_flatten([A_TITLE,'\n',COMP_TEXT,'\n \n'],THE_TEXT).

% Don’t ask me about existent birds though. I own the latest
% edition of the Sibley Guide to Birds, but have only used it to
% identify some kind of swallow that was filling up the nearby trees,
% and one common crane. I took a class on animal diversity and we did a
% unit on birds of paradise - their varying calls, the particular
% excesses of their tails and plumage - but I skipped class that week
% and never got around to making up the reading. I did like birds of
% prey when 8 or 9, but mostly because of their speed and killing. But I
% do stop even when a little late to stare a moment at a thrush in a
% tree or to watch a chicken in the grass, it’s head bouncing up with
% grubs.

ofBird_ofTypes_text(_,[],"").

ofBird_ofTypes_text(LIST_BIRD,[identificatory|OTHER_TYPES],THE_TEXT):-
        identificatory_sentence(LIST_BIRD,ID_SENT),
        ofBird_ofTypes_text(LIST_BIRD,OTHER_TYPES,REMAINING_TEXT),
        string_concat(ID_SENT,REMAINING_TEXT,THE_TEXT).
ofBird_ofTypes_text(LIST_BIRD,[distribution|OTHER_TYPES],THE_TEXT):-
        distributionAndDiet_sentence(LIST_BIRD,DIST_SENT),
        ofBird_ofTypes_text(LIST_BIRD,OTHER_TYPES,REMAINING_TEXT),
        string_concat(DIST_SENT,REMAINING_TEXT,THE_TEXT).
ofBird_ofTypes_text(LIST_BIRD,[descriptive|OTHER_TYPES],THE_TEXT):-
        descriptive_sentence(LIST_BIRD,DESC_SENT),
        ofBird_ofTypes_text(LIST_BIRD,OTHER_TYPES,REMAINING_TEXT),
        string_concat(DESC_SENT,REMAINING_TEXT,THE_TEXT).
ofBird_ofTypes_text(LIST_BIRD,[behaviour|OTHER_TYPES],THE_TEXT):-
        behaviour_sentence(LIST_BIRD,BEH_SENT),
        ofBird_ofTypes_text(LIST_BIRD,OTHER_TYPES,REMAINING_TEXT),
        string_concat(BEH_SENT,REMAINING_TEXT,THE_TEXT).
ofBird_ofTypes_text(LIST_BIRD,[song|OTHER_TYPES],THE_TEXT):-
        song_sentence(LIST_BIRD,SONG_SENT),
        ofBird_ofTypes_text(LIST_BIRD,OTHER_TYPES,REMAINING_TEXT),
        string_concat(SONG_SENT,REMAINING_TEXT,THE_TEXT).
ofBird_ofTypes_text(LIST_BIRD,[nothing|OTHER_TYPES],THE_TEXT):-
        ofBird_ofTypes_text(LIST_BIRD,OTHER_TYPES,REMAINING_TEXT),
        birdList_birdName(LIST_BIRD, HAS_A_NAME),
        strs_flatten(["We know nothing about the ", HAS_A_NAME,"."], NOTHING_SENT),
        string_concat(NOTHING_SENT,REMAINING_TEXT,THE_TEXT).


% I did airplanes though - the two engined, the four-engined, the now
% mostly extinct three-engined; the curve of nose telling Airbus from
% Boeing; the presence or absence of extra winglets distinguishing A320
% and A340.

desc_text_patterns([[descriptive,identificatory,distribution,behaviour,song],
                    [descriptive,identificatory,distribution,behaviour,song],
                    [descriptive,distribution,behaviour,song]]).

descriptive_text(LISTED_BIRD,THE_TEXT):-
        birdList_birdName(LISTED_BIRD,A_NAME),
        string_upper(A_NAME,A_TITLE),
        desc_text_patterns(DESC_TEXT_PATTERNS),
        random_member(DESC_TEXT_PATTERN,DESC_TEXT_PATTERNS),
        ofBird_ofTypes_text(LISTED_BIRD,DESC_TEXT_PATTERN,DESC_TEXT),
        strs_flatten([A_TITLE,’\n’,DESC_TEXT,’\n \n’],THE_TEXT).

% Caged Bird
% BY MAYA ANGELOU
%
% A free bird leaps
% on the back of the wind
% and floats downstream
% till the current ends
% and dips his wing
% in the orange sun rays
% and dares to claim the sky.
%
% But a bird that stalks
% down his narrow cage
% can seldom see through
% his bars of rage
% his wings are clipped and
% his feet are tied
% so he opens his throat to sing.
%
% The caged bird sings
% with a fearful trill
% of things unknown
% but longed for still
% and his tune is heard
% on the distant hill
% for the caged bird
% sings of freedom.
%
% The free bird thinks of another breeze
% and the trade winds soft through the sighing trees
% and the fat worms waiting on a dawn bright lawn
% and he names the sky his own
%
% But a caged bird stands on the grave of dreams
% his shadow shouts on a nightmare scream
% his wings are clipped and his feet are tied
% so he opens his throat to sing.
%
% The caged bird sings
% with a fearful trill
% of things unknown
% but longed for still
% and his tune is heard
% on the distant hill
% for the caged bird
% sings of freedom.

find_birds(0,[],_,[],_).

find_birds(HOW_MANY,[],INDEX,BIRDS_FOUND,WORDS_FOUND):-
        HOW_MANY >= 1,
        INDEX >= 1,
        a_birdListNum(FIRST_BIRD,INDEX),
        descriptive_text(FIRST_BIRD,FIRST_WORDS),
        ONE_LESS is HOW_MANY - 1,
        NEXT_INDEX is INDEX + 1,
        find_birds(ONE_LESS,[],NEXT_INDEX,OTHER_BIRDS,OTHER_WORDS),
        append(OTHER_WORDS,[FIRST_WORDS],WORDS_FOUND),
        append(OTHER_BIRDS,[FIRST_BIRD],BIRDS_FOUND).

% Something that i think i am particularly troubled by is a sense of
% mechanicity - not so much that I am a mechanichal thing (as my
% language is, as my mathematics and metaphor and desiring), but that i
% am a particularly small and poorly made one; a few rusting pulleys
% roped together and held up by tape and cardboard. Birds strike me as
% the sorts of machines - for watching from trees, for flapping and odd
% song - that i could accept being.

find_birds(HOW_MANY,[FIRST_OLD_BIRD|OTHER_OLD_BIRDS],INDEX,BIRDS_FOUND,WORDS_FOUND):-
        INDEX >= 1,
        a_relationListNum(FIRST_OLD_BIRD,FIRST_BIRD,INDEX),
        NEXT_INDEX is INDEX + 1,
        find_birds(HOW_MANY,OTHER_OLD_BIRDS,NEXT_INDEX,OTHER_BIRDS,OTHER_WORDS),
        comparative_text([FIRST_BIRD,FIRST_OLD_BIRD],NEW_WORDS),
        append(OTHER_WORDS,[NEW_WORDS],WORDS_FOUND),
        append(OTHER_BIRDS,[FIRST_BIRD],BIRDS_FOUND).

% Surfin' Bird
% THE TRASHMEN
%
% A-well-a, everybody's heard about the bird
% Bird, bird, bird, b-bird's the word
% A-well-a, bird, bird, bird, the bird is the word
% A-well-a, bird, bird, bird, well, the bird is the word
% A-well-a, bird, bird, bird, b-bird's the word
% A-well-a, bird, bird, bird, well, the bird is the word
% A-well-a, bird, bird, b-bird's the word
% A-well-a, bird, bird, bird, b-bird's the word
% A-well-a, bird, bird, bird, well, the bird is the word
% A-well-a, bird, bird, b-bird's the word
% A-well-a, don't you know about the bird
% Well, everybody knows that the bird is the word
% A-well-a, bird, bird, b-bird's the word
% A-well-a
% A-well-a, everybody's heard about the bird
% Bird, bird, bird, b-bird's the word
% A-well-a, bird, bird, bird, b-bird's the word
% A-well-a, bird, bird, bird, b-bird's the word
% A-well-a, bird, bird, b-bird's the word
% A-well-a, bird, bird, bird, b-bird's the word
% A-well-a, bird, bird, bird, b-bird's the word
% A-well-a, bird, bird, bird, b-bird's the word
% A-well-a, bird, bird, bird, b-bird's the word
% A-well-a, don't you know about the bird
% Well, everybody's talking about the bird
% A-well-a, bird, bird, b-bird's the word
% A-well-a, bird
% Surfin' bird
% Bbbbbbbbbbbbbbbbbb, aaah
% Pa-pa-pa-pa-pa-pa-pa-pa-pa-pa-pa-pa-pa-pa-pa-pa
% Pa-pa-pa-pa-pa-pa-pa-pa-pa-pa-pa-pa-pa-pa-ooma-mow-mow
% Papa-ooma-mow-mow
% Papa-ooma-mow-mow, papa-ooma-mow-mow
% Papa-ooma-mow-mow, papa-ooma-mow-mow
% Ooma-mow-mow, papa-ooma-mow-mow
% Papa-ooma-mow-mow, papa-ooma-mow-mow
% Papa-ooma-mow-mow, papa-ooma-mow-mow
% Oom-oom-oom-oom-ooma-mow-mow
% Papa-ooma-mow-mow, papa-oom-oom-oom
% Oom-ooma-mow-mow, papa-ooma-mow-mow
% Ooma-mow-mow, papa-ooma-mow-mow
% Papa-a-mow-mow, papa-ooma-mow-mow
% Papa-ooma-mow-mow, ooma-mow-mow
% Papa-ooma-mow-mow, ooma-mow-mow
% Papa-oom-oom-oom-oom-ooma-mow-mow
% Oom-oom-oom-oom-ooma-mow-mow
% Ooma-mow-mow, papa-ooma-mow-mow
% Papa-ooma-mow-mow, ooma-mow-mow
% Well, don't you know about the bird
% Well, everybody knows that the bird is the word
% A-well-a, bird, bird, b-bird's the word
% Papa-ooma-mow-mow, papa-ooma-mow-mow
% Papa-ooma-mow-mow, papa-ooma-mow-mow
% Papa-ooma-mow-mow, papa-ooma-mow-mow...

fibonacci_birds(0,[],_,"").

fibonacci_birds(1,BIRD,BIRD,WORD):-
        find_birds(1,[],1,BIRD,WORD).

fibonacci_birds(N,NEW_BIRDS,BIRDS,WORDS):-
        N >= 2,
        NMONE is N - 1,
        NMTWO is N - 2,
        fibonacci(N,FIB_N),
        fibonacci(NMTWO,FIB_NMTWO),
        fibonacci_birds(NMONE,RECENT_BIRDS,OLD_BIRDS,OLD_WORDS),
        find_birds(FIB_NMTWO,RECENT_BIRDS,FIB_N,NEW_BIRDS,NEW_WORDS),
        append(OLD_BIRDS,NEW_BIRDS,BIRDS),
        append(OLD_WORDS,["- \n\n"],PREV_WORDS),
        append(PREV_WORDS,NEW_WORDS,WORDS).

% Beginning early with an odd bird in the hand, one free in the sheet of
% the sky, or two on the wire.
%
%-
%
% What i am writing about is that i go home, and the next morning we go
% on a hike up the local public park mountain and what we do is carry a
% laminated guide to local wildflowers and at each new blossom stop and
% cluster around the pictures looking at the yellow white purple orange
% or red petals comparing and looking at leaf lengths and saying into
% the wind "California Milkworth," "Purple Larkspur," "Baby Blue Eyes,"
% "Mugwort," "Morning Glory," "Fiddleneck," "Yellow Monkeyflower,"
% "Linseed"
%
%-
```

```
%
% In the blurb of Inger Christensen’s /Alphabet/, someone calls her a
% "singer of syllables." That’s one thing i believe in i guess, the
% saying of the names of things as a kind of prayer or assurance of your
% place among them and towards no other end, to speak in clicking
% noun-phrases and breathing infinitives while walking through both the
% landscape and the words for it.
%
% There is that classic sort of wizardry that operates by knowing and
% speaking, as distinguished from simply recalling and recreating the
% sounds of, names. From Ursula Le Guin’s /Earthsea/ for instance "My
% name, and yours, and the true name of the sun, or a spring of water,
% or an unborn child, all are syllables of the great word that is very
% slowly spoken by the shining of the stars. There is no other power. No
% other name." Something else i like about this kind of Young Adult
% wizardry is the suspicion of the power in it: how the knowledge of a
% person’s actual name gives a total and dangerous kind of control over
% them; the namer of birds, there, is a maker of lists but also
% responsible for the unpredictable and at times incendiary or
% entrapping results of speaking their grimoire.
%
%-
%
% Code is language that does fixed things.
%
% Code is not what this is about but it is one of the things that it is
% about because, for all of programming’s silliness obscurity and
% economics, in its sentences one can glimpse every now and then,
% through the For loops and cautious architectures of parenthesis,
% inside the obscure formal spacing and odd capitalization, traces of
% the old original magic: words, utterances that some opaque and not
% human thing hears and responds to.
%
% The rhetoric of almost all programming is the rhetoric of either the
% imperative or of the declarative: in the first the programmer tells
% their variables what they are, what they will do, what they will
% mutate into, and eventually how they will be written or acted upon;
% functional programming describes how one sort of code-thing makes
% another code-thing, and writes the rules for these subtle growths and
% transformations until one finds themselves writing rules about rules,
% and rules about rules about rules. Logic programming however, as this
% document mostly is, is an attempt at a third approach: in Prolog one
% describes a system of things, and if this constructed place is a
% possible one the interpreter finds it, assembles the pieces from
% basic blocks, and shows you the discovered landscape.
%
%-
%
% Consider an eccentric, aspiring, and trust-fund supported
% ornithologist who absolutely refuses to step out of their room.
%
% Birds occasionally fly past their window or rest on a distant branch,
% but each too quickly or at just too much of a distance for the
% Ornithologist At Their Window to do more than observe one feature or a
% rough outline of.
%
% This eccentric however feels for whatever reason compelled to sit for
% hours daily at their desk with a pen and a notebook, assembling these
% remembered fragments into descriptions of the birds which they imagine
% they might have seen. They might tell the way a blue neck inflates and
% contracts over indigo wings, or the three syllable screeching of
% certain hawks. It is important for this story that the only way we can
% write about this author is to talk about how they write their
% ornithology; much later, someone finds the enormous stack of notebooks
% they leave behind containing nothing but potential birds. The contents
% of their notebooks are published on someone else’s whim.
%
% Now imagine walking outside with this guide, down to the slough in
% Spring or into hills, and seeing a bird on a branch of an oak, or
% wading through the muck.
%
% You open the Book of Potential Birds and flip for a while through
% its exhaustive pages, learning to navigate its oddly ordered
% sections. The bird is in no rush to get anywhere else, has a long
% stripe accross its torso, and hobbles and bobs along. Now you find an
% entry in the book that - though its author never saw the bird in front
% of you, describes exactly its size, motion, and one stripe. The book
% tells you the bird makes a harsh and doubled whistling; this time,
% the bird you are watching a body’s length away makes a harsh and
% doubled whistling.
%
%-
%
% There are birds in the backyard: after metaphor, fact and background
% noise.
%
% Over fig blossoms a green one is humming, still as the landscape moves
% around it.
%
% On the fence at the back of the property, seen through the leaves of
% the apricot tree are three gray and brown animals, to my untrained
% eyes the only description for their shape "bird": a doubly bent curve,
% simple beak, a bulge and suggestion of wings, the thin toes wrapped
% round wood and the paper fan of tail behind. Their heads and torsos
% turn occasionally and at once; they strut along the walk. They
% pause. Then, they fly up at my approach and land back three feet
% further down along the wood.
%
% Two sleek and brown-chested specimens pick at the fallen apricots or
% the grubs buried in the fruits opened orange matter.
%
% One small and dark and gray thrashes around somehow within air, as if
% touching and thrown back by invisible walls which its wings find and
% shove.
%
% A long way up two bent flecks, making somewhere small running shadows,
% drift.
%
% Another bird with a dark crest, white neck, neatly splayed tail,
% the kind of bird for the birdwatchers or the hikers at their cameras,
% rests a few moments in another long-leaved tree, before setting the
% branch vibrating as the blur of it dives up and swims in air.
%
% The background noise, as it almost invariably is, is a range of bird
% whistles, chirps, and chittering - the apparent peaks each with their
% own jagged and private contour - that stand against each other; now a
% wail that comes in threes, rising, dropping, while from another ear’s
% side of the yard a full paragraph of evenly spaced and paper-thin
% notes scatters, with an undertone of insects vibrating and, if you
% listen very carefully, the refrigerator.
%
% -
%
% There is a 13 paragraph story-shaped hole here, and what i am doing is
% looking for the noises or meanings that could succesfully fill out or
% scaffold it; reaching for birds: consider a landscape the birds have
% deserted - electric wires just strings sagging their symmetric natural
% sag, braches and leaves that jiggle predictably to wind, fences which
% designate boundaries and have nothing to interrupt their tops.
%
% The topography of this birdless place is the same as it was, more or
% less, but the place somehow seems empty as a page, as though it was
% existing before to hold birds, as though while watching this landscape
% you are really waiting for the birds and their utterances to inhabit
% it, and are now looking for where they will enter it again: on the
% branches, along the fence, or clutching the wire.
%
% Maybe birds are parralel and alternate societies, civilizations we
% look into to restructure or make contingent ours: buildings of rooks,
% and coveys of partridges; murmurations of starlings, casts of hawks,
% and the inevitable murders of crows; some parliaments of owls, and a
% watch of nightingales. To sit in the parliamentary chamber observing
% unceasing arguments in an unspoken language.
%
```

```
% Alternately, maybe single birds are instead examples of the
% metamorphosed bodies an inacessible person becomes: the understanding
% of herons and cruelty of ravens; wisdom of owls, grace of swans, and
% morbid despair of vultures. The way the animal cannot protest the
% story told of it without its call being made back into story, or to
% praise.
%
% Something else i like and look for is the moment when a metaphor gets
% so involved in its details that it no longer appears to be a map to
% somewhere else, but its own place - parables that get lost in
% themselves, like Aesop’s Fables or Calvino’s /Cosmicomics/; i believe
% in maps, in the possibility of paraphrase - that, at the end of the
% day, our explanations usually can do what we need them to - but also
% that there is something else to be found in the contour of the drawn
% coastline, a way in which the pencil finding the inlets and little
% islands also begins to undertand the possible joy and reasons behind
% its own motion. I think one could read The Sibley Guide to Birds,
% as a magic-realist collection of prose-poetry now, if there were no
% birds around at all.
%
% ’Consider a language-game between a birdwatcher A and a
% nature photographer B. A is photographing birds: there are grebes,
% thrushes, swallows, and wrens. B has to find and name the birds in the
% order in which A wants to photograph them. For this purpose, they use
% a language consisting of the words "grebe," "thrush," "swallow," and
% "wren." A calls them out; B finds and points to the bird they have
% learned to find at such-and-such a call. Concieve this as a complete
% primitive language.’ Consider a sendentary creature who returns
% through all seasons to the same three or four perches (i’m trying to
% avoid autobiography, but only find myself explaining what i imagine i
% do and do not understand of myself).
%
% Raymond Queneau?s motto: "Rats who build the labyrinth from which they
% will try to escape"
%
% Georges Perec’s: "I set myself rules in order to be totally free"
%
% This is not the logic of birds of course, but the logic of a mind
% making a logic for the birds already in its private landscape: i do
% not hope to find the actual patterns of ornithology, but only to
% consider some of the odd extravagant and particular birds of a logic;
% it is not the still polygon of the stuffed specimen but the blur of
% the hovering hummingbird’s wings that i want to watch and understand:
% or maybe what is interesting is actually the counterpoint between
% polygon and motion.
%
% I do remember walking through the coastal forest where an osprey was
% and the binoculars on a string around my neck. I do not really
% remember the bird beyond the flash of some section of white and the
% signification of wings, but i remember talking and reading about it
% afterwards, in our place a rare bird, fish-eater and diver, nesting
% on large sticks placed in a dead tree. I remember kestrels on
% telephone wires and boring vultures. A hawk or swift that one morning
% we found after a thunderclap from the living room window: shaking in a
% pile there for some minutes as we wondered whether there was someone
% one is supposed to call about dying birds, when it gathered itself up
% and flapped off.
%
% A speculation (not true or untrue but perhaps with the sound of
% potentially resonant things, that can catch and keep an ear for the
% duration of their utterance and which later a mouth may find itself
% repeating a variation on): birdsong is the place where names and
% things line up; where, for the length of such a song, the landscape
% and its language can be figure and ground of the same place; a
% rabbit running from the hawk’s shadow, the hawk directing the
% darkening of grass after the rabbit.
%
% You are taking a break now, finding a window, finding the bird in it,
% giving it a minute, coming back after.
%
% "Maybe sparrow it’s too late / Moonlight glanced off metal wings / In
% a thunderstorm above the clouds / The engine hums a sparrow’s phrase
% / For those who cannot hear the words / For those who will not hear
% the words / For those who will not hear the words / La di da di da di
% da / La di da di da di da" - Neko Case

lots_ofBirdWords(N,THE_TEXT):-
        get_time(NOW),
% "Lightning -
        fibonacci_birds(N,_,_,THE_WORDS),
% Heron's cry
        append([["The Guide to Nonexistent Birds:",
                 " an Ornithological Logic",
                 "\n\nSECTION ",NOW,":\n\n",
                 "* * ** *** ***** ******** *************",
                 "\n\n"],
                THE_WORDS],ALL_WORDS),
        strs_flatten(ALL_WORDS,THE_TEXT).
% Stabs the darkness" - BASHO (trans. unknown)

guideToNonexistentBirds(BIRD_WORDS):-
        lots_ofBirdWords(7,BIRD_WORDS).

% Freebird
% LYNRYD SKYNYRD
%
% If I leave here tomorrow
% Would you still remember me?
% For I must be traveling on now
% 'Cause there's too many places I've got to see.
%
% But if I stayed here with you, girl,
% Things just couldn't be the same.
% 'Cause I'm as free as a bird now,
% And this bird you can not change, oh, oh, oh, oh.
% And this bird you can not change.
% And this bird you can not change.
% Lord knows I can't change.
%
% Bye, bye, baby, it's been a sweet love, yeah,
% Though this feeling I can't change.
% But please don't take it so badly,
% 'Cause Lord knows I'm to blame.
%
% But if I stayed here with you, girl,
% Things just couldn't be the same.
% 'Cause I'm as free as a bird now,
% And this bird you'll never change, oh, oh, oh, oh.
% And this bird you cannot change.
% And this bird you cannot change.
% Lord knows, I can't change.
% Lord, help me, I can't change.
%
% Lord, I can't change.
% Won't you fly high, free bird, yeah?

say_birds(_Request) :-
    guideToNonexistentBirds(BIRD_WORDS),
    reply_html_page(
            [title('The Guide to Nonexistent Birds: an Ornithological Logic')],
            [
            pre(style="white-space: pre-wrap; width:45%; height:100%; float: right", BIRD_WORDS),
            iframe([src='https://titleduntitled.name/facades/OrnithologicalLogic.pdf', style='wid
th:50%; height:95%; float: left; margin: 4px 10px 0px 0px; position: fixed'],[])]).
```