

Draft Proposal Version 0.1: Simple python builtins in Congress for Datalog expressions:
Prabhakar Kudva

```
//Example from Tim  
// +, *, and lt are builtins  
p(x, y, z) :- q(x), r(z) (3*x)+2=y, lt(y, z)
```

Only prefix notations will be supported initially. Additional library functions will be added where they don't exist in native python.

Category: comparison

lt(x,y), gt, ge, le, eq

Category: string manipulation

concat(string1, string2)

len(string1, lenstring1) // output is lenstring1

in(string1, regularexpression)

Category: Arithmetic

arithmetic: +, -, *, /, <, >, <=, >=

type-casting: from number to string and from string to number

Checks and safety for all literals that are predicates in the body:

1. Builtins may only appear in the body of a rule
2. Each builtin has certain arguments deemed 'inputs' and other arguments deemed 'outputs'
3. Safety with builtins is a bit hard to state b/c it is recursive. A builtin is safe if all of its input variables appear in a non-builtin or as an output of a safe builtin.
4. A term in a 'builtin predicate' can only exist if it also exists in a 'relational predicate' in the body of the clause, or is part of the head
5. Will allow nested and recursive definitions, given 'Relational predicates' cannot be nested with builtin predicates, and builtin predicates are not nested in relational.
6. Builtin predicate symbols are restricted to the ones in the library
7. Every term in a builtin literal should be reducible to a constant before passing to a python eval
8. The builtins cannot contain any of the datalog punctuation characters

Implementation:

Compiler will separate the builtins from the remaining literals in the body. All terms that are variables are instantiated to their values by deduction. Once all terms in a builtin predicate are reduced to constants, a python eval is performed on the expression/function. The returned value is then used by runtime.py along with the values of the other literals in the body to reduce the RHS.

A *reduce_builtin_args* function will be implemented.

Installing new builtins:

```
register_congress_listing_category(category_name)
register_congress_builtin (name, args)
register_taxonomy ( listing_category, array(name), attribute map)
```

The data structures to maintain builtins:

A dictionary of builtins will be maintained by category of the form:

```
[ category: list_of_builtins ]
builtin = [ name: 'name', arg_list: [list_of_args], call_func: python_function_name ]
```

[1] <http://www.ccs.neu.edu/home/ramsdell/tools/datalog/datalog.html#Tutorial>

[2] <http://courses.cs.washington.edu/courses/cse544/00sp/lectures/ppt/l8.ppt#282,18,Deductive Databases>

[3] <http://cse.unl.edu/~riedesel/pub/cse413/des/doc/manualDES.pdf>

[4] <http://infolab.stanford.edu/~ullman/cs345notes/cs345-1.ppt#269,14,Evaluating Datalog Programs>

[5] <http://infolab.stanford.edu/~ullman/cs345notes/cs345-1.ppt#275,20,Example>

[6] <http://www-users.cselabs.umn.edu/classes/Spring-2010/csci8211/Readings/datalog.pdf>

[7] <https://pypi.python.org/pypi/pyDatalog>