

PatternOperations package

Mark A. Caprio, Center for Theoretical Physics, Yale University

Version 1.0 (July 8, 2005)

Introduction

Patterns, or generic expressions which can be matched by a variety specific expressions, are central to symbolic computation, programming, and list processing in *Mathematica*. Some patterns are straightforward to define in *Mathematica*, but others which *should* be are not. For example, `_Integer` is a pattern which represents any integer, so selecting all integers from a list is accomplished easily, as

```
Cases[{1, 2, 7.35, Pi}, _Integer]
{1, 2}
```

But to instead define a pattern which matches any *noninteger* requires extraordinary gymnastics, for instance, as

```
Cases[{1, 2, 7.35, Pi}, x_ /; !MatchQ[x, _Integer]]
{7.35, Pi}
```

The PatternOperations package provides functions which simplify the construction of more complicated patterns. These functions allow patterns to be modified or combined through "logic" operations analogous to the ordinary logic functions Not, And, Or, *etc.* For instance the above selection of nonintegers can be carried out easily, simply by negating the `_Integer` pattern, as

```
Cases[{1, 2, 7.35, Pi}, PatternNot[_Integer]]
{7.35, Pi}
```

Pattern operations

<code>PatternNot[p]</code>	Pattern matched by anything not matching <i>p</i> .
<code>PatternAnd[p1, p2, ...]</code>	Pattern matched by anything matching <i>p1</i> and <i>p2</i> , <i>etc.</i>
<code>PatternOr[p1, p2, ...]</code>	Pattern matched by anything matching <i>p1</i> or <i>p2</i> , <i>etc.</i>
<code>PatternNand[p1, p2, ...]</code>	Pattern matched by anything <i>not</i> matching <i>p1</i> and <i>p2</i> , <i>etc.</i>
<code>PatternNor[p1, p2, ...]</code>	Pattern matched by anything <i>not</i> matching <i>p1</i> or <i>p2</i> , <i>etc.</i>
<code>PatternXor[p1, p2, ...]</code>	Pattern matched by anything matching <i>p1</i> or <i>p2</i> but not both. See <code>Xor</code> help for behavior with more than two arguments.
<code>PatternImplies[p1, p2]</code>	Pattern matched by anything <i>not</i> matching <i>p1</i> or matching <i>p2</i> .

Pattern logic operations.

<code>PatternTrue</code>	Pattern matched by anything, analogous to the logical value <code>True</code> . Equivalent to <code>Blank[]</code> .
<code>PatternFalse</code>	Pattern matched by nothing, analogous to the logical value <code>False</code> .

Pattern constants.

The most useful pattern operations are `PatternNot`, `PatternAnd`, `PatternOr`, and perhaps `PatternNor`. The rest are provided for completeness, in analogy to the standard logic operations. Some further examples follow.

Select explicit integers or reals but nothing else

```
Cases[{1, 2, 7.35, Pi, 1 + 5 * I}, PatternOr[_Integer, _Real]]
{1, 2, 7.35}
```

Select anything except explicit integers or reals

```
Cases[{1, 2, 7.35, Pi, 1 + 5 * I}, PatternNot[PatternOr[_Integer, _Real]]]
{π, 1 + 5 i}

Cases[{1, 2, 7.35, Pi, 1 + 5 * I}, PatternNor[_Integer, _Real]]
{π, 1 + 5 i}
```

Select composite integers (positive, not unity, and not prime) not greater than 100

```
Cases[
  {-8, 1, 5, 12, 49, 999, Pi},
  PatternAnd[_Integer, _?Positive, PatternNot[1], PatternNot[_?PrimeQ], x_ /; (x ≤ 100)]
]
{12, 49}

Cases[
  Range[0, 100],
  PatternAnd[_Integer, _?Positive, PatternNot[1], PatternNot[_?PrimeQ], x_ /; (x ≤ 100)]
]
{4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28,
 30, 32, 33, 34, 35, 36, 38, 39, 40, 42, 44, 45, 46, 48, 49, 50, 51, 52, 54,
 55, 56, 57, 58, 60, 62, 63, 64, 65, 66, 68, 69, 70, 72, 74, 75, 76, 77, 78,
 80, 81, 82, 84, 85, 86, 87, 88, 90, 91, 92, 93, 94, 95, 96, 98, 99, 100}
```

© Copyright 2005, Mark A. Caprio.