XPath for XML Navigation

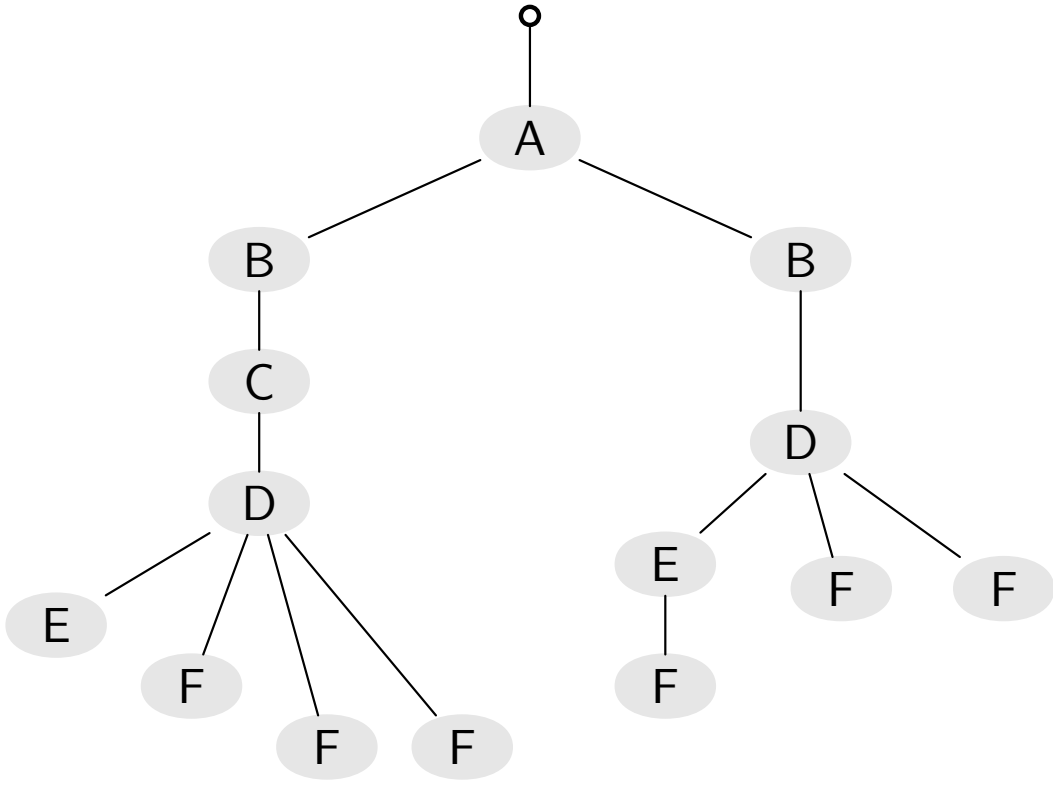
Janis Voigtländer

Technische Universität Dresden

Summer Term 2009

XPath for Tree Navigation: An Example

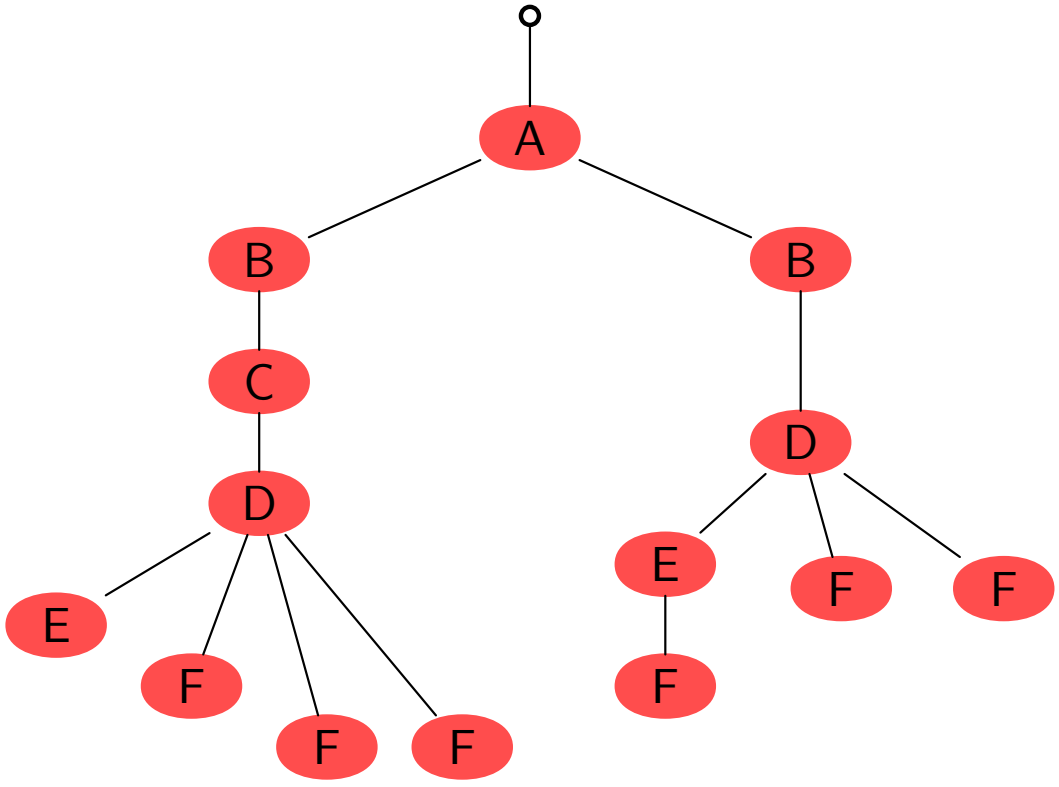
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

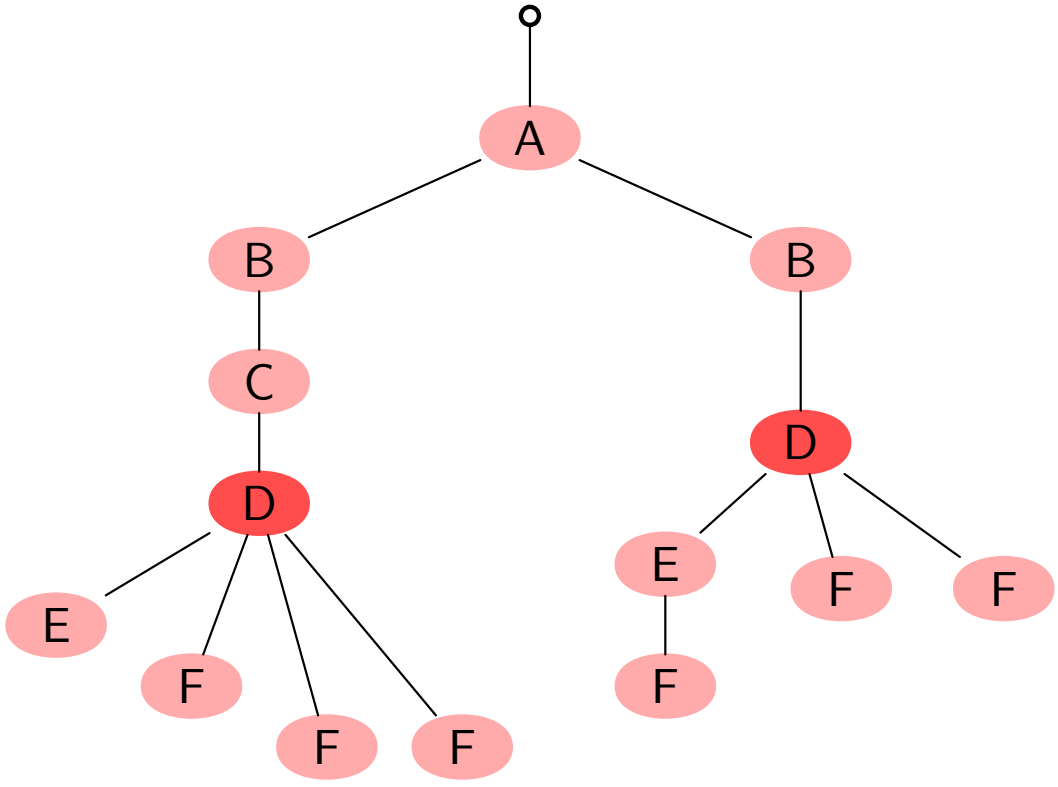
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

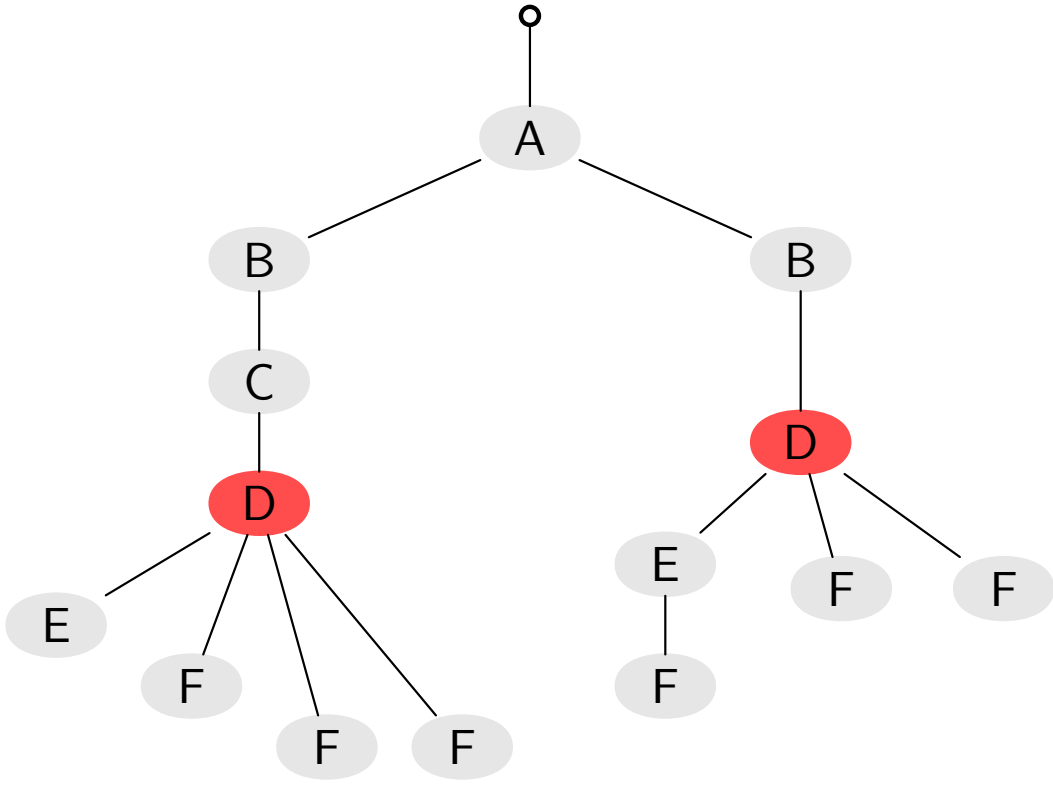
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

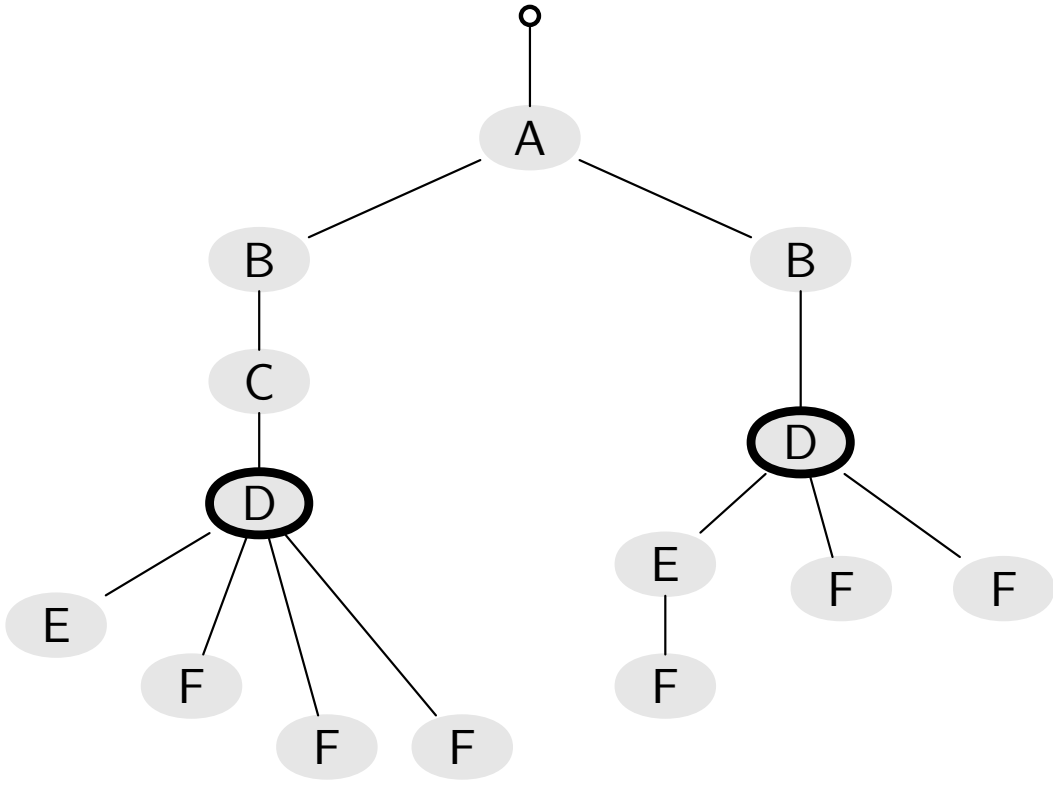
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

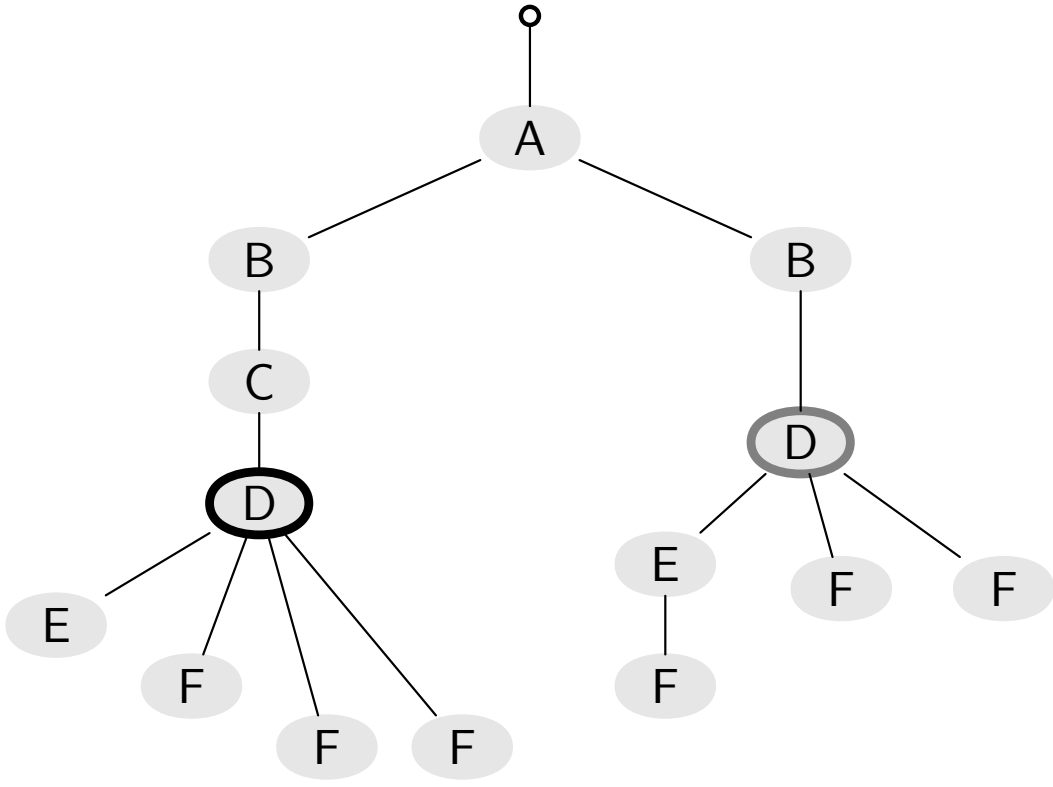
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

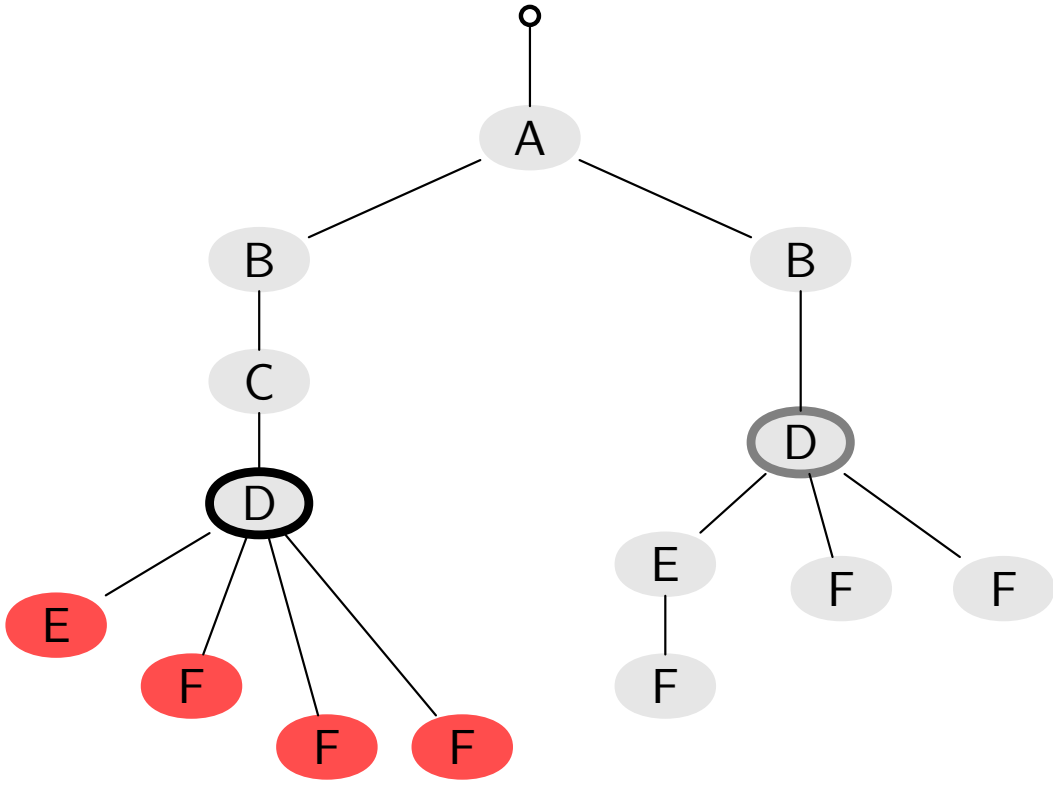
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

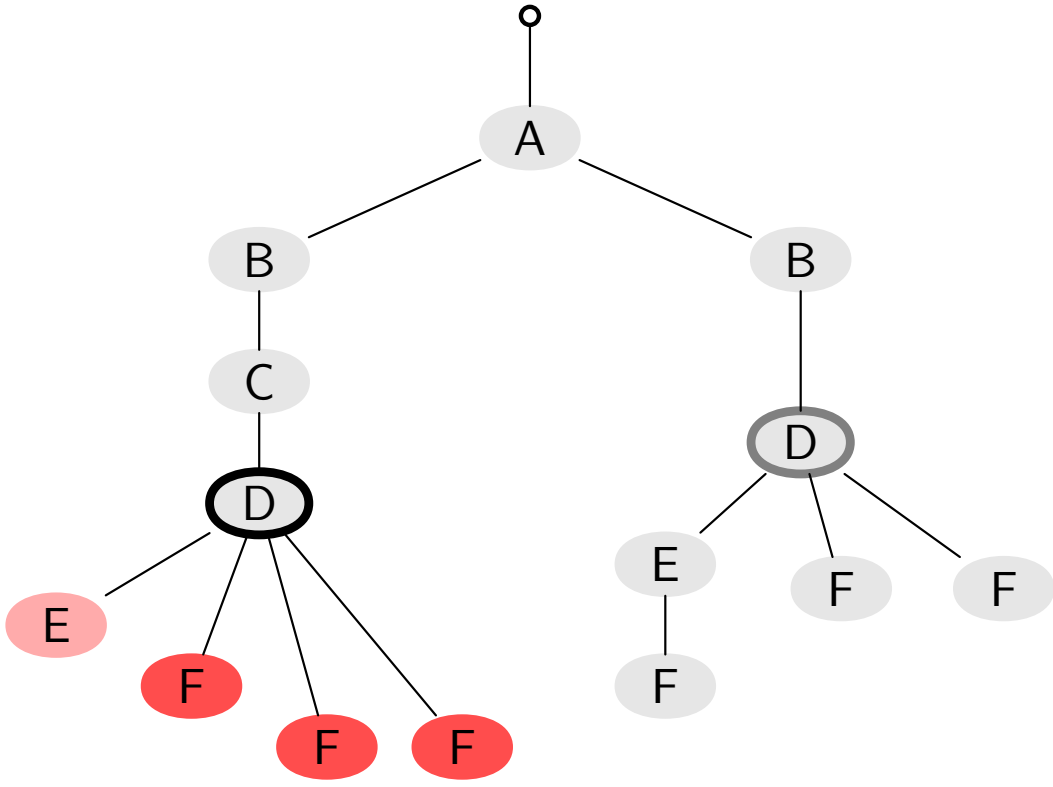
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

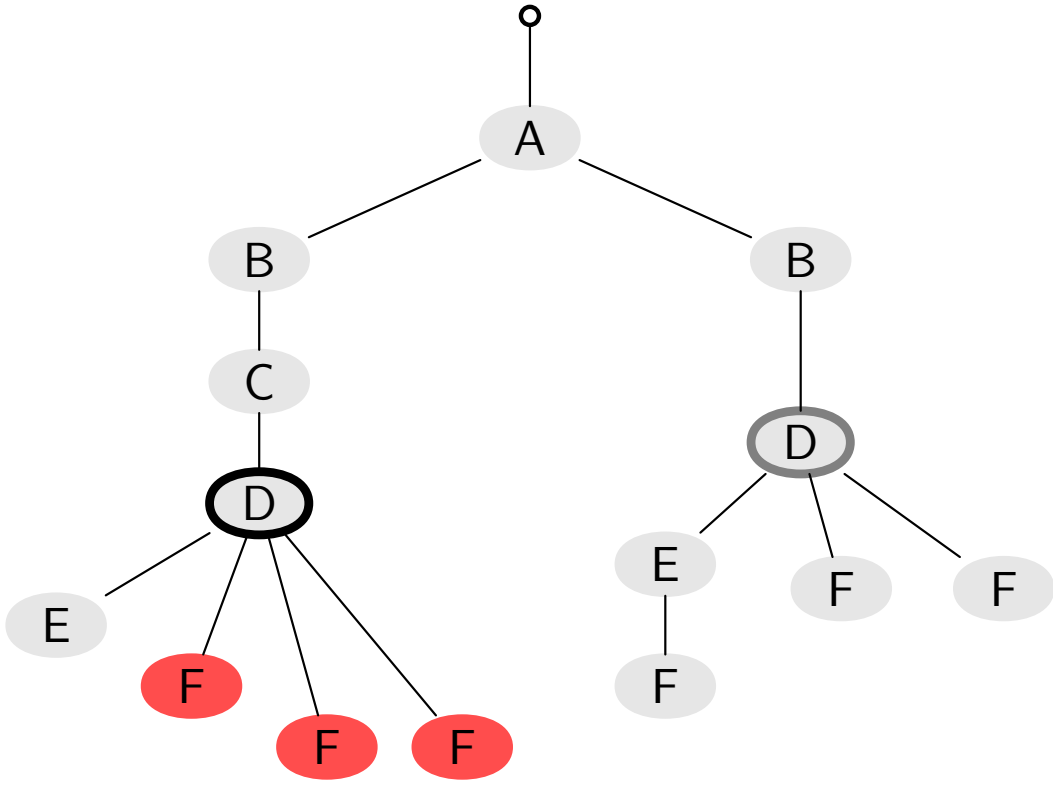
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

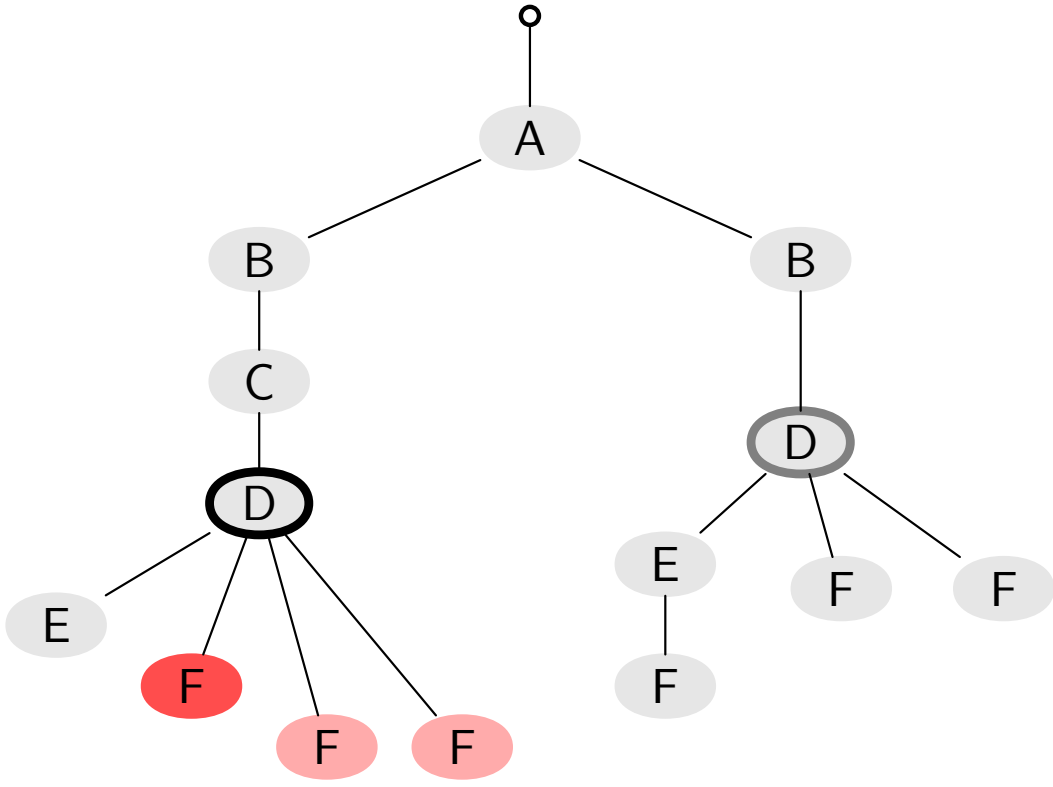
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

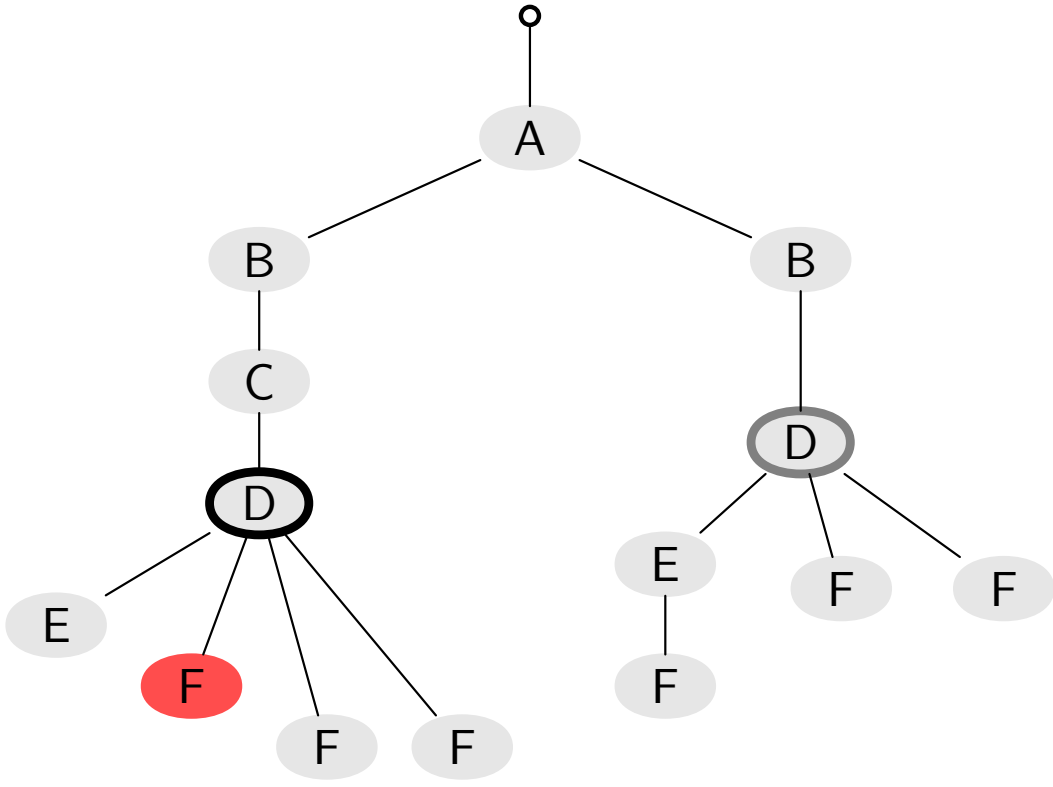
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

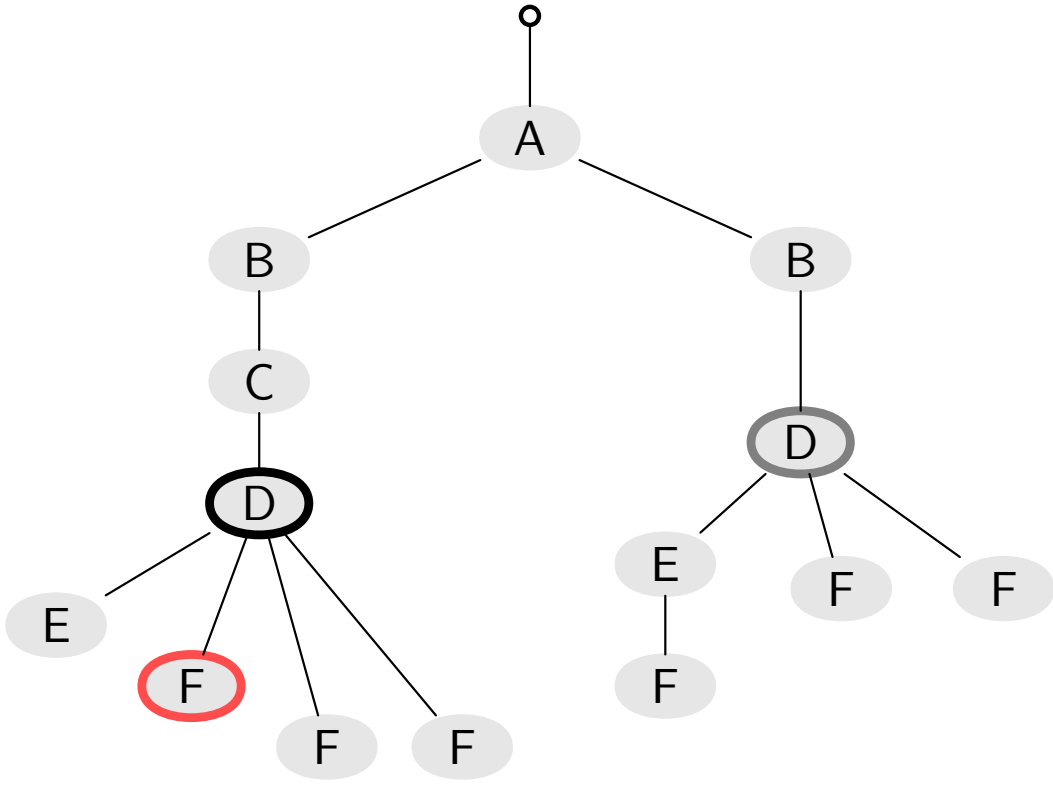
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

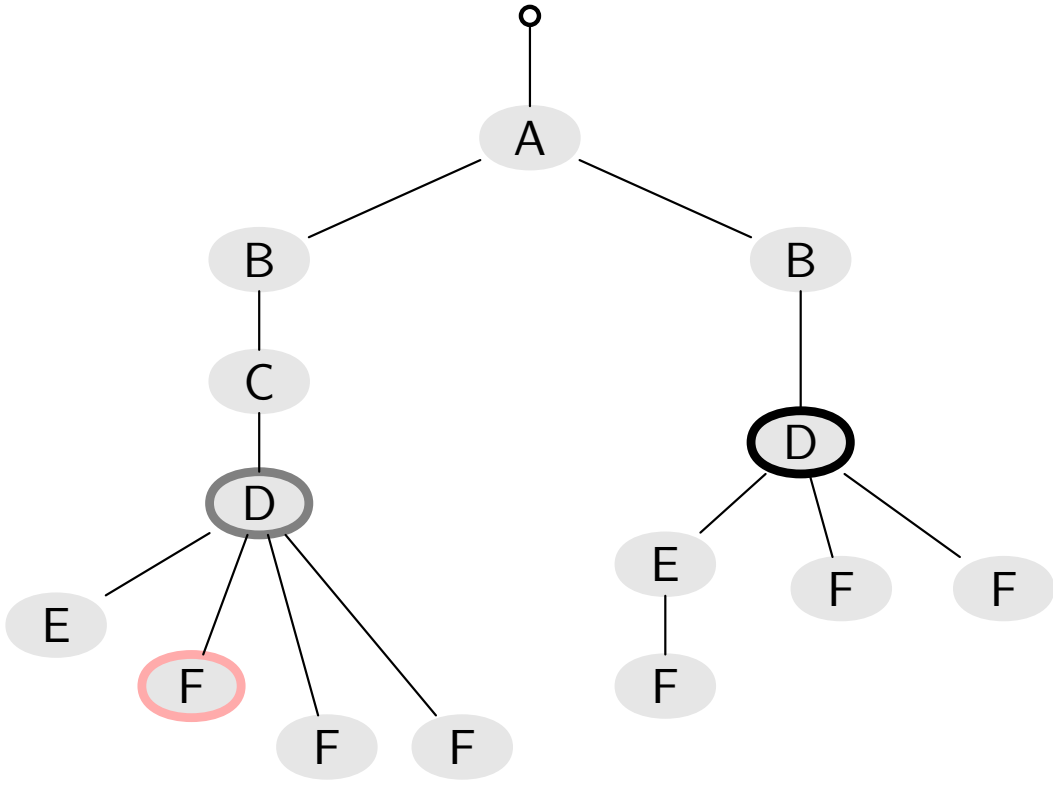
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

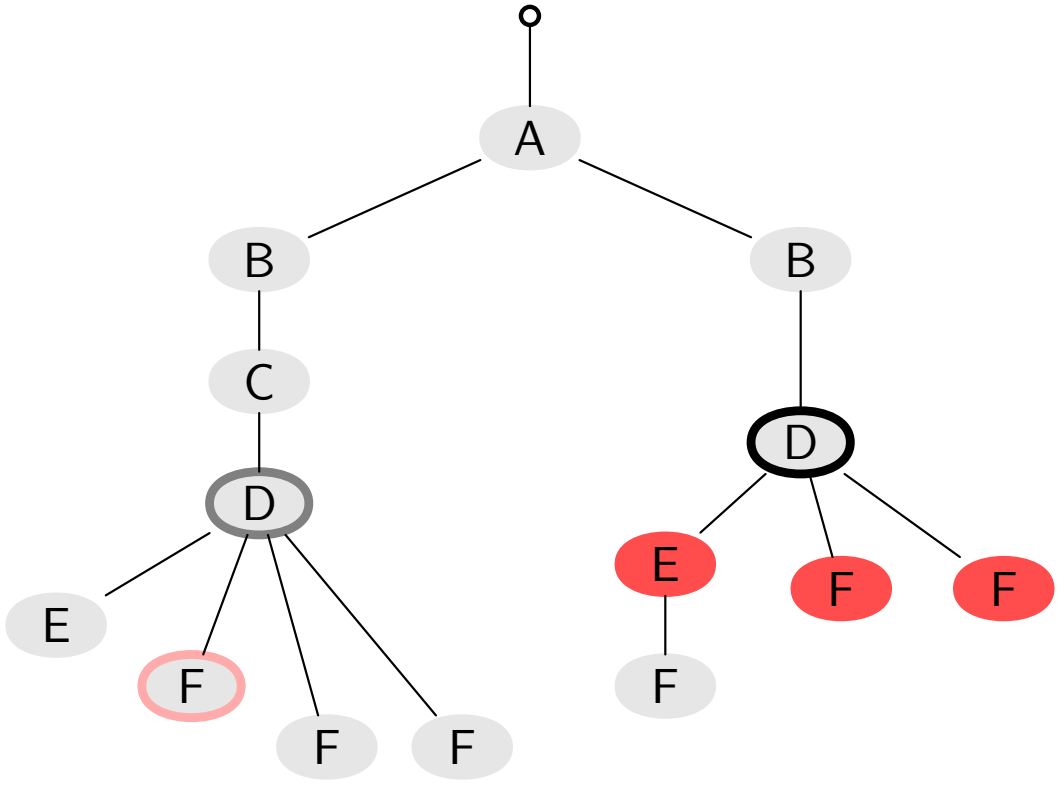
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

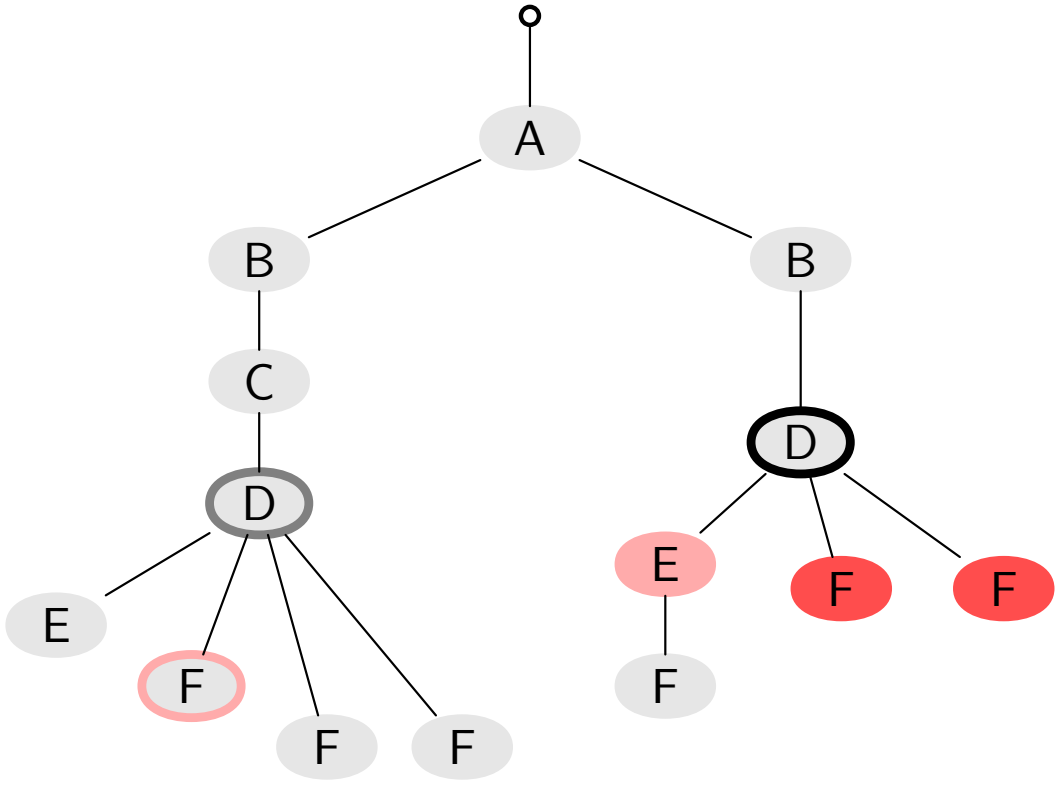
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

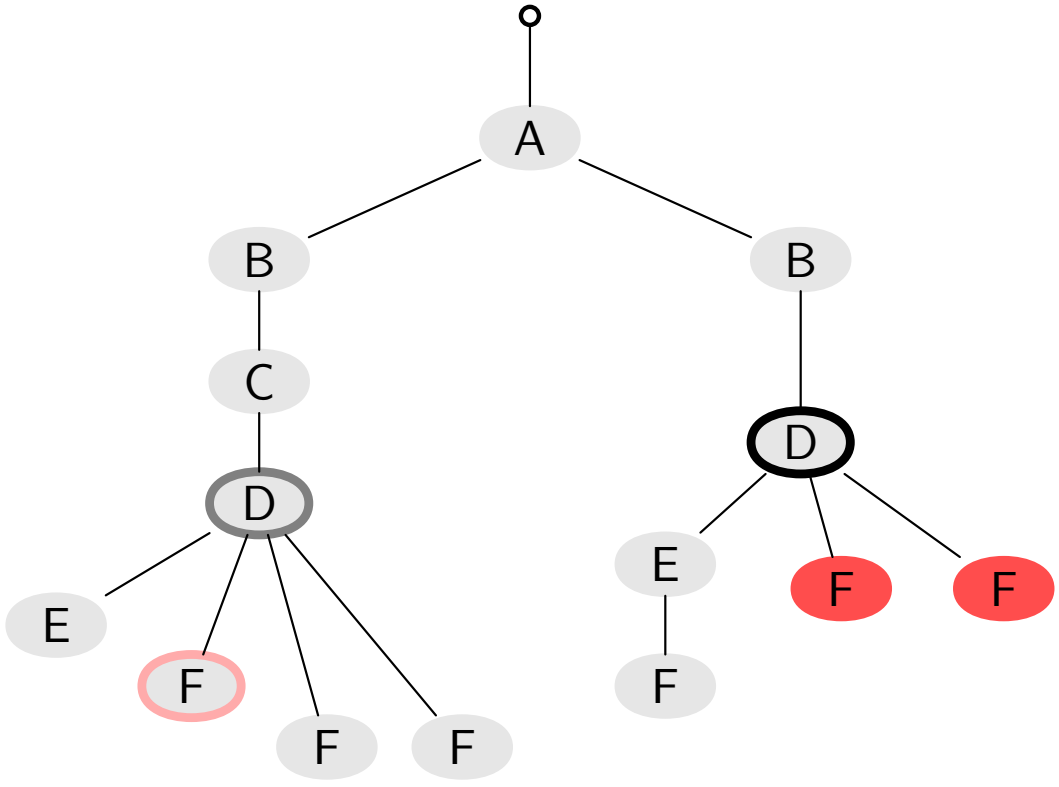
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

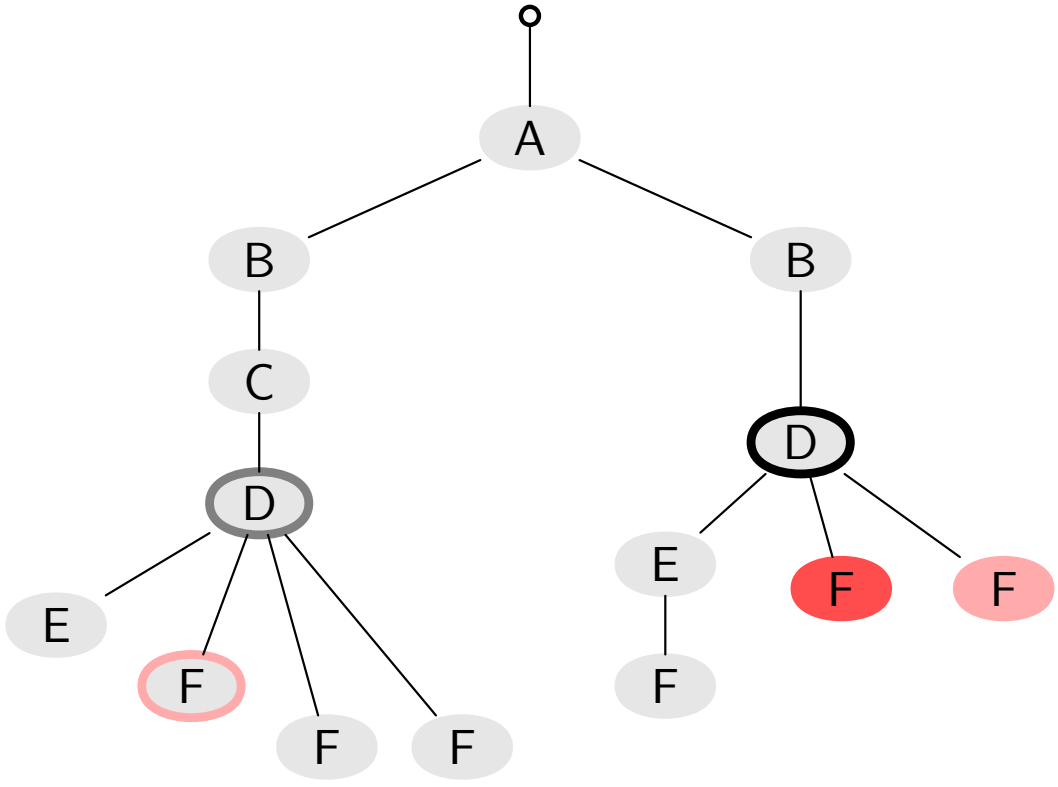
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

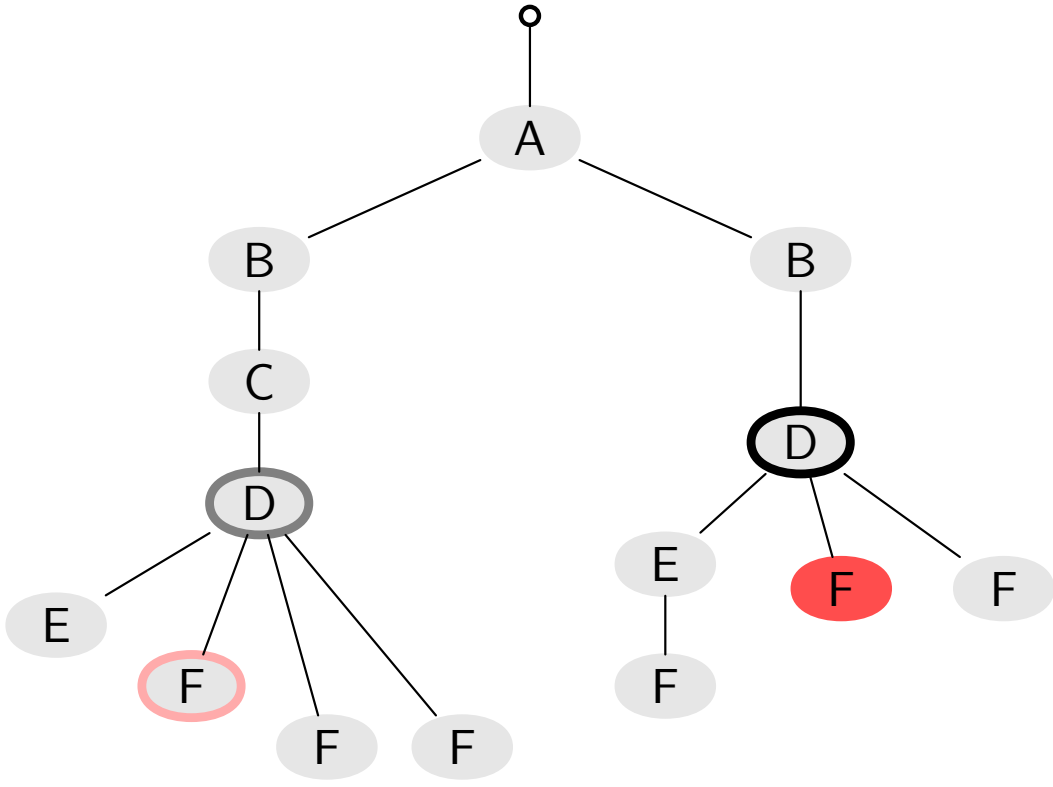
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

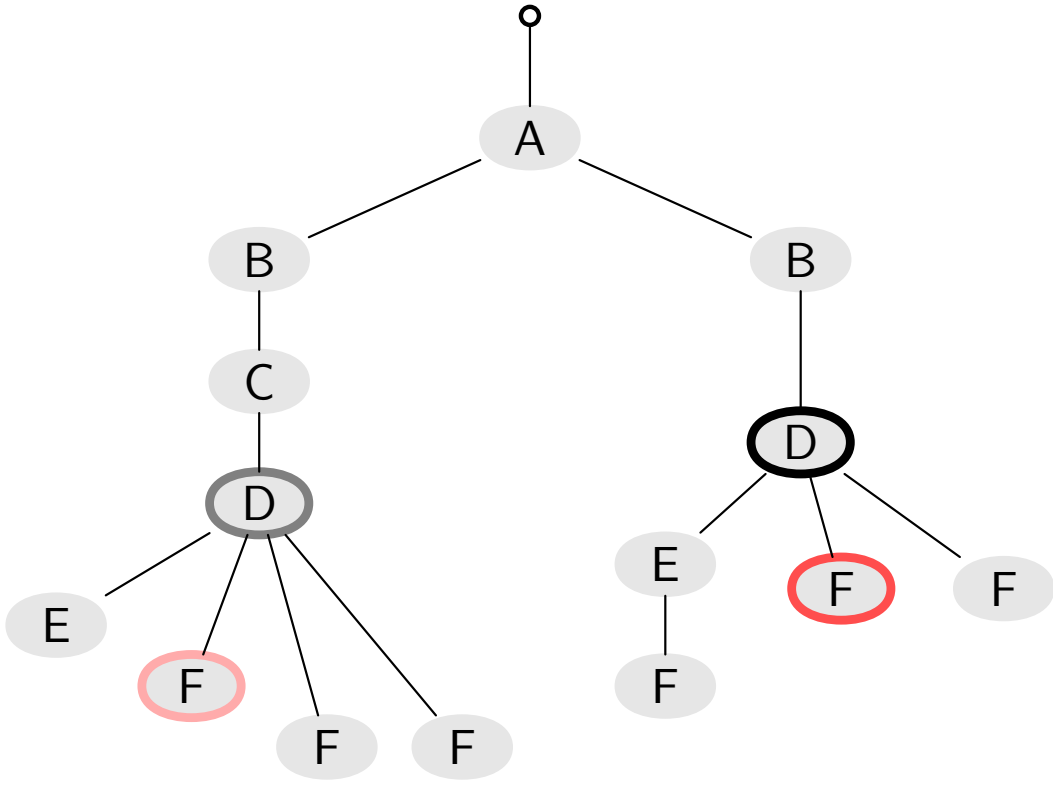
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

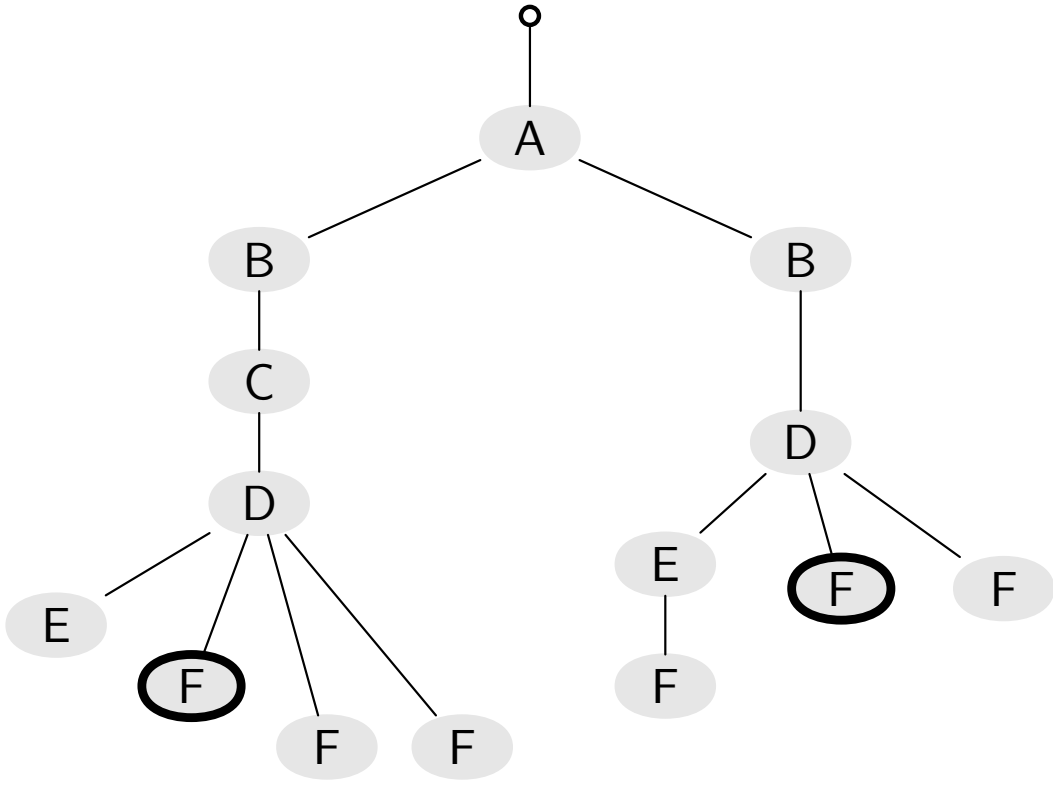
Document:



Query: `/descendant::D/child::F[position()=1]`

XPath for Tree Navigation: An Example

Document:



Query: `/descendant::D/child::F[position()=1]`

XPath Expressions

Paths: `/descendant::D/child::F[position()=1]`

XPath Expressions

Paths: `/descendant::D/child::F[position()=1]`

Steps: `descendant::D`

XPath Expressions

Paths: `/descendant::D/child::F[position()=1]`

Steps:
```
descendant::D
child::F[position()=1]
```

XPath Expressions

Paths: `/descendant::D/child::F[position()=1]`

Steps: `descendant::D`

`child::F[position()=1]`

Axes: `descendant`, `child`, ...

XPath Expressions

Paths: `/descendant::D/child::F[position()=1]`

Steps: `descendant::D`

`child::F[position()=1]`

Axes: `descendant`, `child`, ...

Tests: `D`, `F`, ...

XPath Expressions

Paths: `/descendant::D/child::F[position()=1]`

Steps: `descendant::D`

`child::F[position()=1]`

Axes: `descendant`, `child`, ...

Tests: `D`, `F`, ...

Filters: `position()=1`, ...

XPath Expressions

Paths: `/descendant::D/child::F[position()=1]`

Steps: `descendant::D`

`child::F[position()=1]`

Axes: `descendant`, `child`, ...

Tests: `D`, `F`, ...

Filters: `position()=1`, ...

General syntax:

$$path \ ::= \ step/\cdots/step \ \mid \ /step/\cdots/step$$

XPath Expressions

Paths: `/descendant::D/child::F[position()=1]`

Steps: `descendant::D`

`child::F[position()=1]`

Axes: `descendant`, `child`, ...

Tests: `D`, `F`, ...

Filters: `position()=1`, ...

General syntax:

$$\begin{aligned} \mathit{path} &::= \mathit{step}/\cdots/\mathit{step} \mid /\mathit{step}/\cdots/\mathit{step} \\ \mathit{step} &::= \mathit{axis}::\mathit{test}[\mathit{fexp}]\cdots[\mathit{fexp}] \end{aligned}$$

XPath Expressions

Paths: `/descendant::D/child::F[position()=1]`

Steps: `descendant::D`

`child::F[position()=1]`

Axes: `descendant`, `child`, ...

Tests: `D`, `F`, ...

Filters: `position()=1`, ...

General syntax:

$$
\begin{array}{lcl}
\mathit{path} & ::= & \mathit{step}/\cdots/\mathit{step} \;\mid\; /\mathit{step}/\cdots/\mathit{step} \\
\mathit{step} & ::= & \mathit{axis} :: \mathit{test}[\mathit{fexp}]\cdots[\mathit{fexp}] \\
\mathit{test} & ::= & \texttt{name} \mid \texttt{node()} \mid \texttt{text()} \mid \cdots
\end{array}
$$

XPath Expressions

Paths: `/descendant::D/child::F[position()=1]`

Steps: `descendant::D`

`child::F[position()=1]`

Axes: `descendant`, `child`, ...

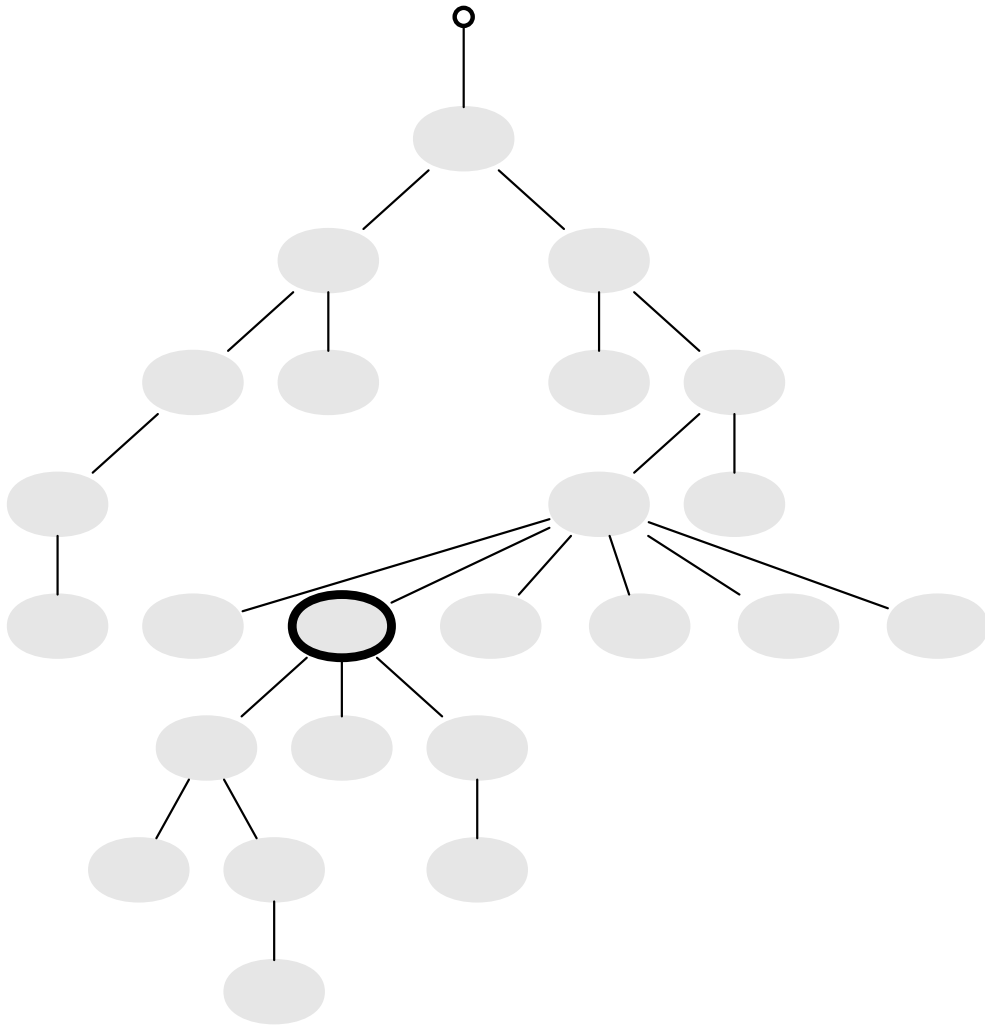
Tests: `D`, `F`, ...

Filters: `position()=1`, ...

General syntax:

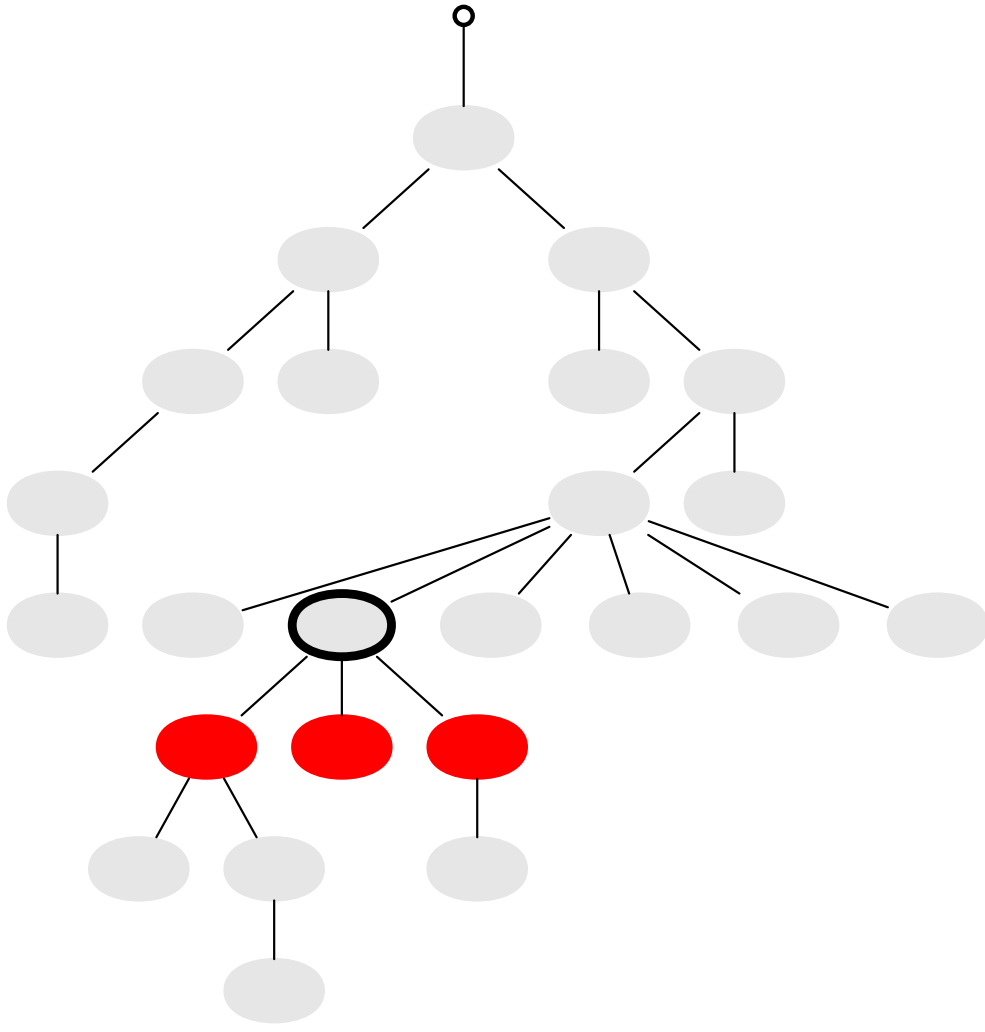
$$
\begin{array}{lcl}
\mathit{path} & ::= & \mathit{step}/\cdots/\mathit{step} \;\mid\; /\mathit{step}/\cdots/\mathit{step} \\
\mathit{step} & ::= & \mathit{axis} :: \mathit{test}[\mathit{fexp}]\cdots[\mathit{fexp}] \\
\mathit{test} & ::= & \texttt{name} \mid \texttt{node()} \mid \texttt{text()} \mid \cdots \\
\mathit{fexp} & ::= & \cdots
\end{array}
$$

Axes, from a Context Node



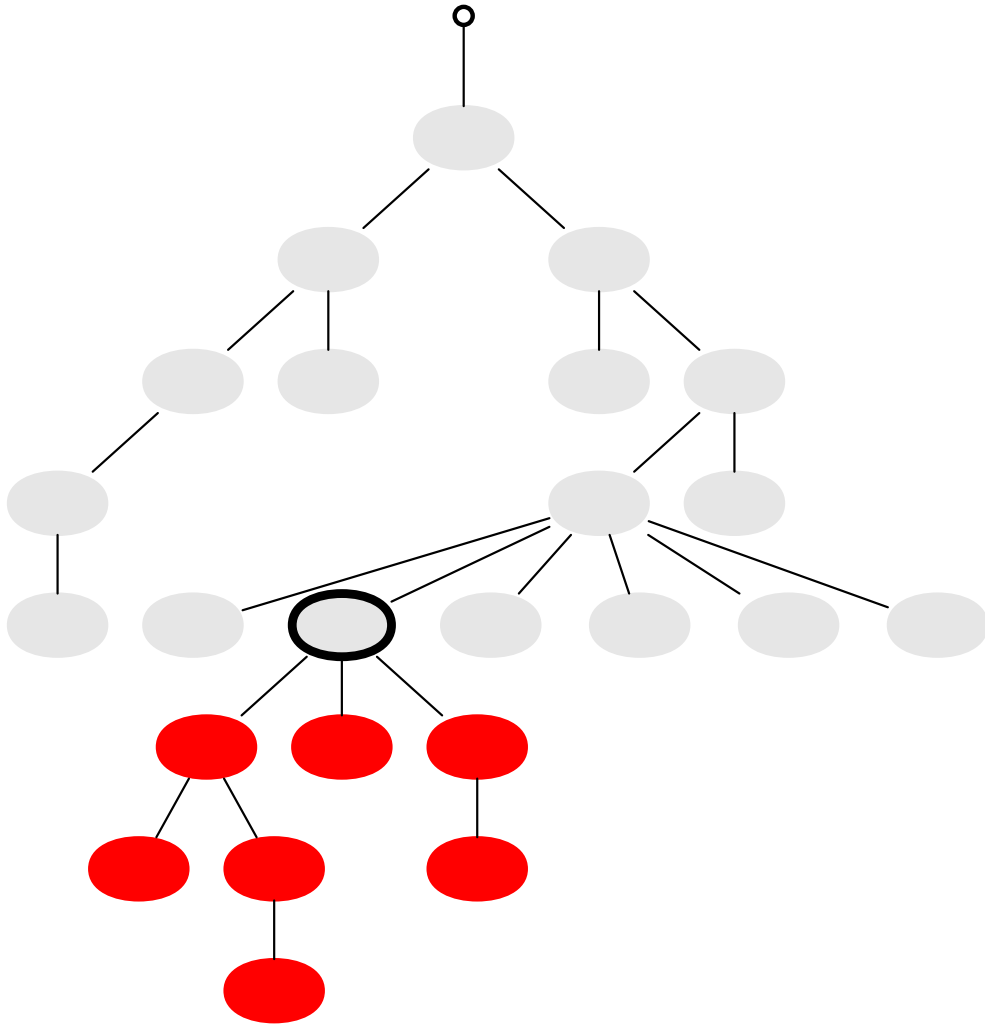
Axes, from a Context Node

`child`



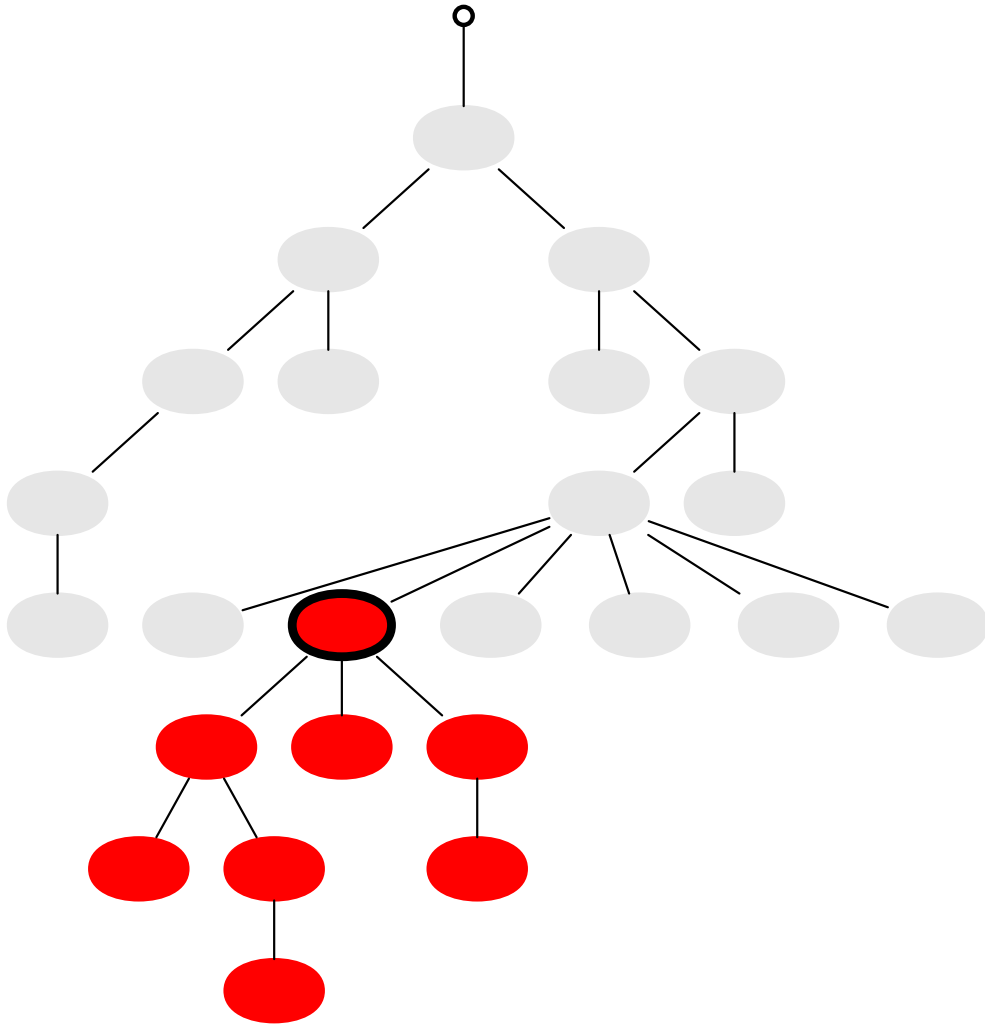
Axes, from a Context Node

`descendant`



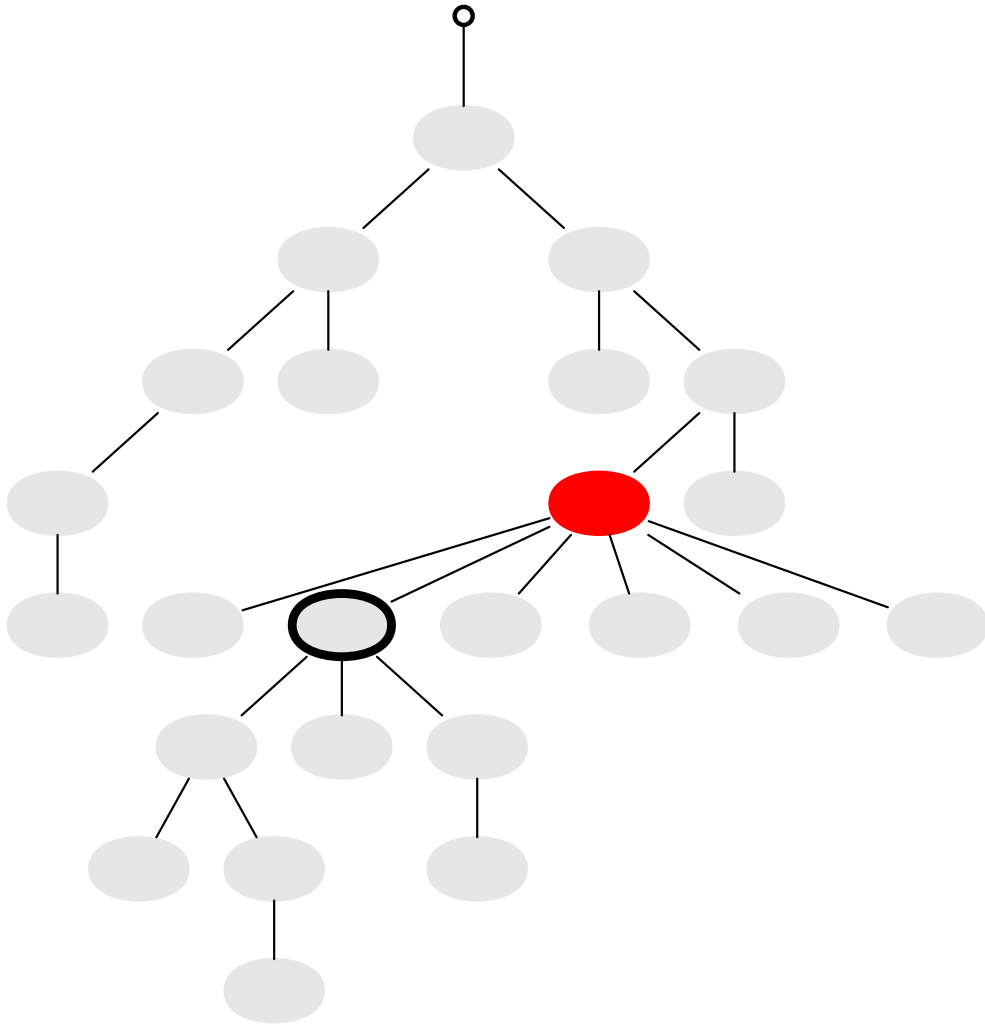
Axes, from a Context Node

`descendant-or-self`



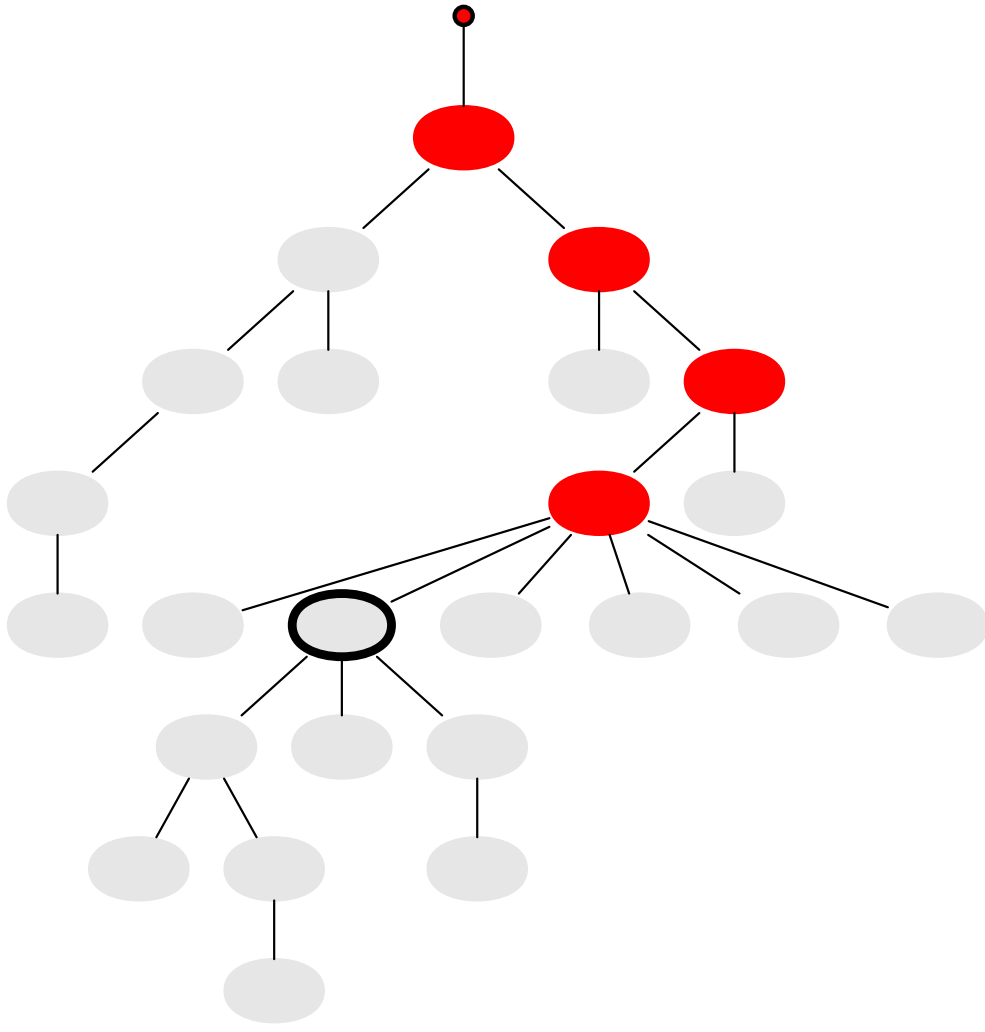
Axes, from a Context Node

`parent`



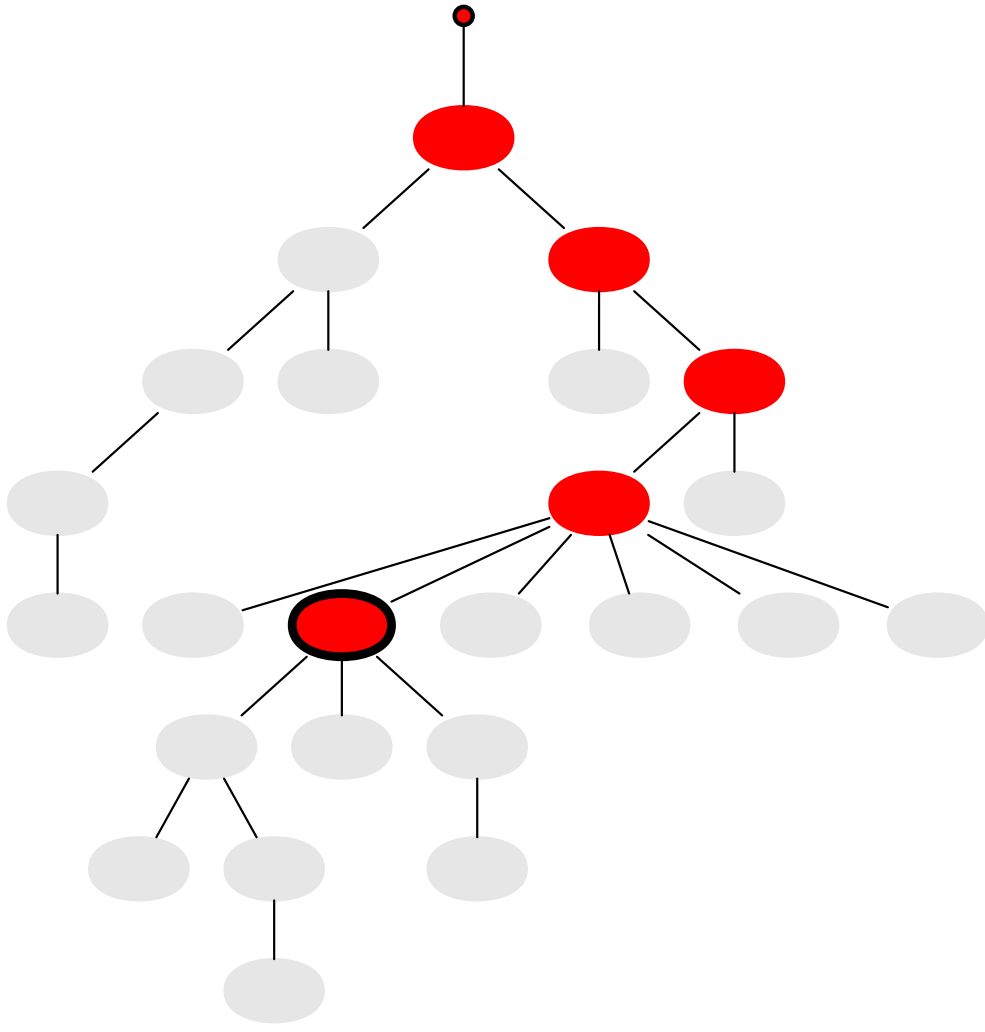
Axes, from a Context Node

`ancestor`



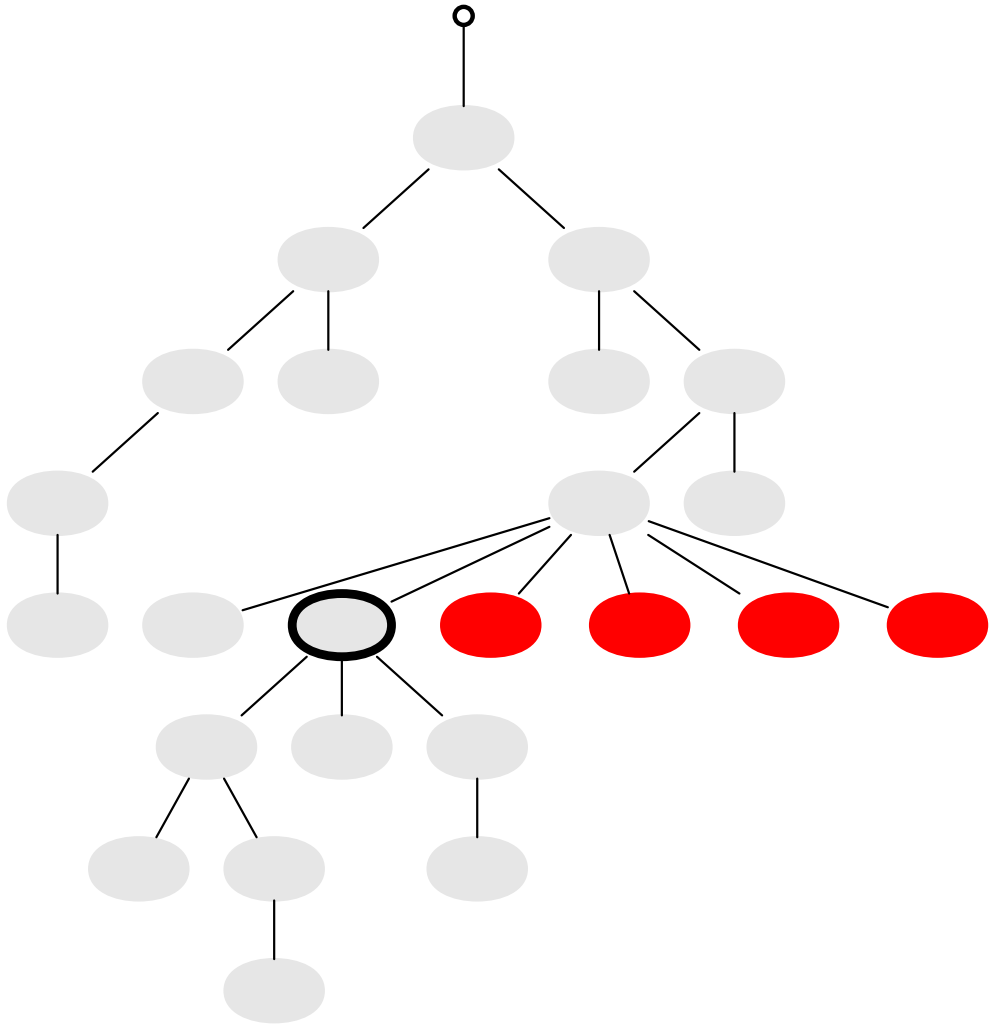
Axes, from a Context Node

`ancestor-or-self`



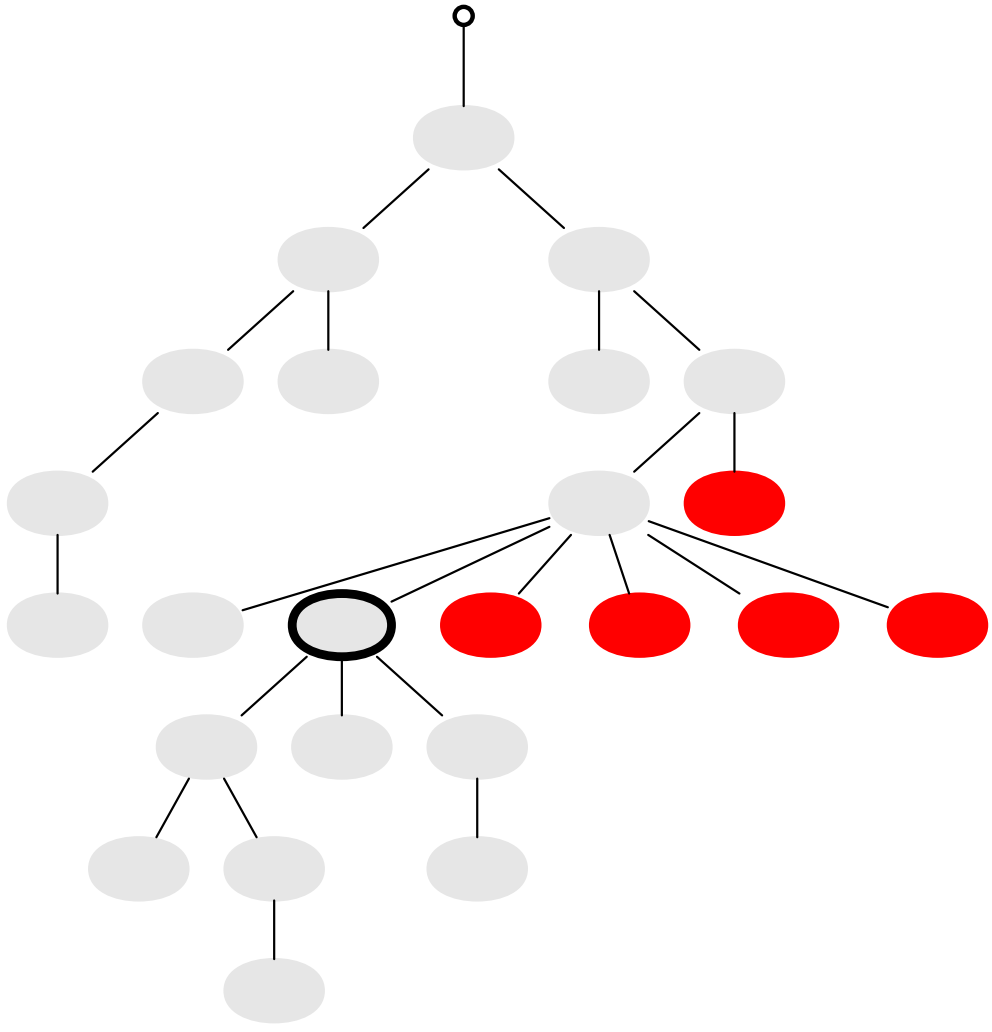
Axes, from a Context Node

`following-sibling`



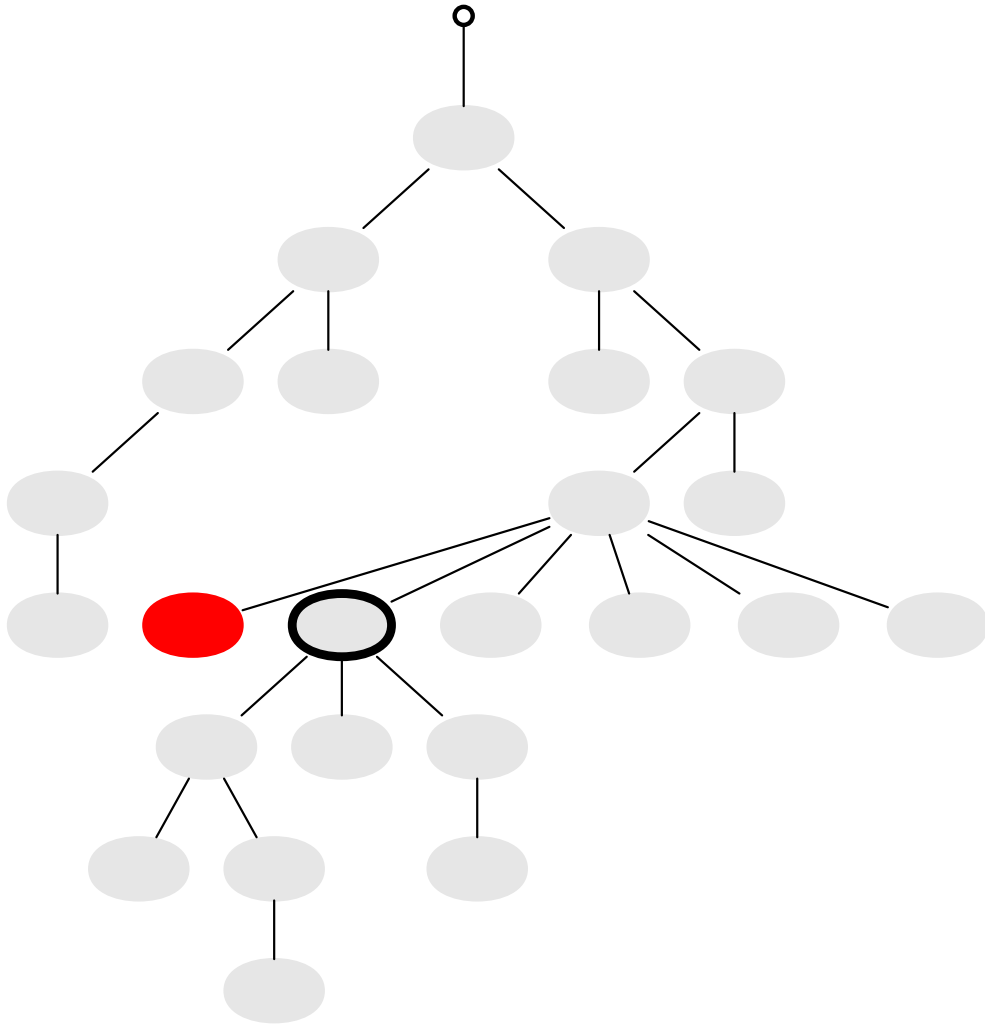
Axes, from a Context Node

`following`



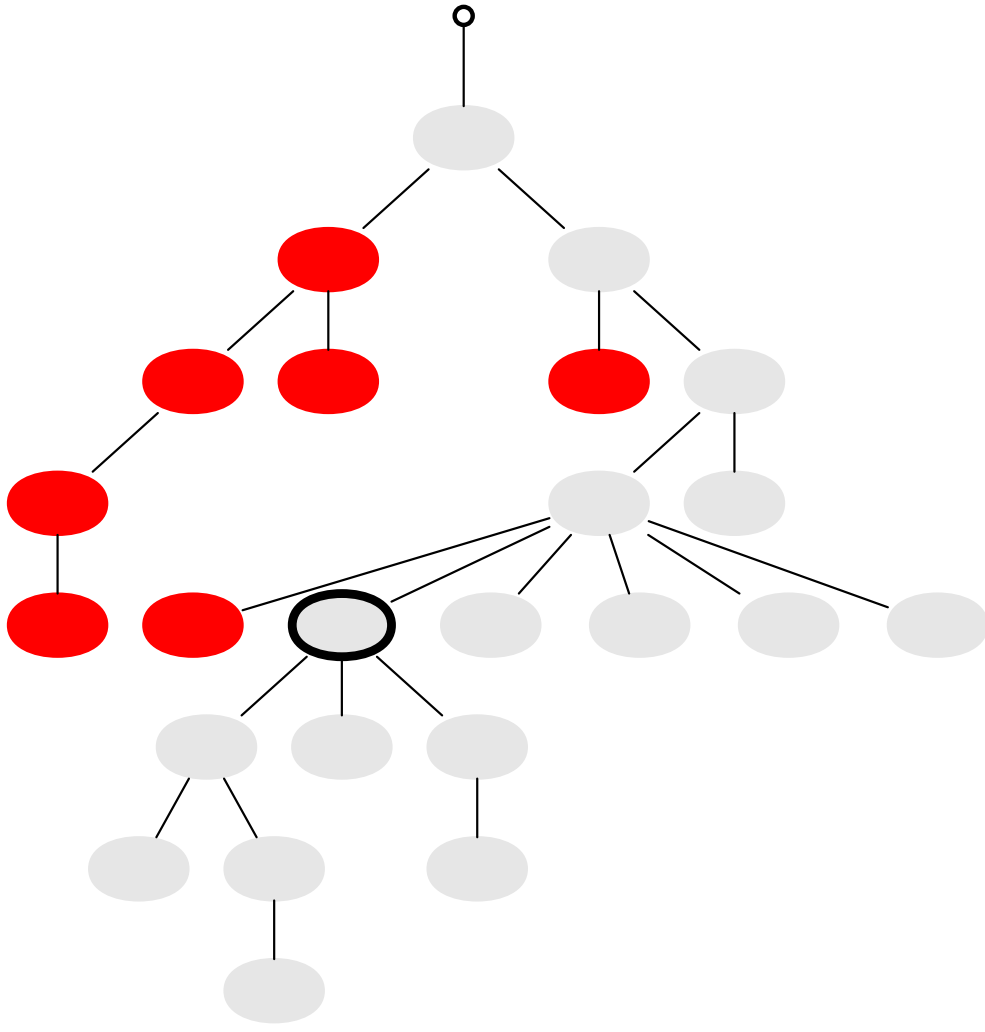
Axes, from a Context Node

`preceding-sibling`



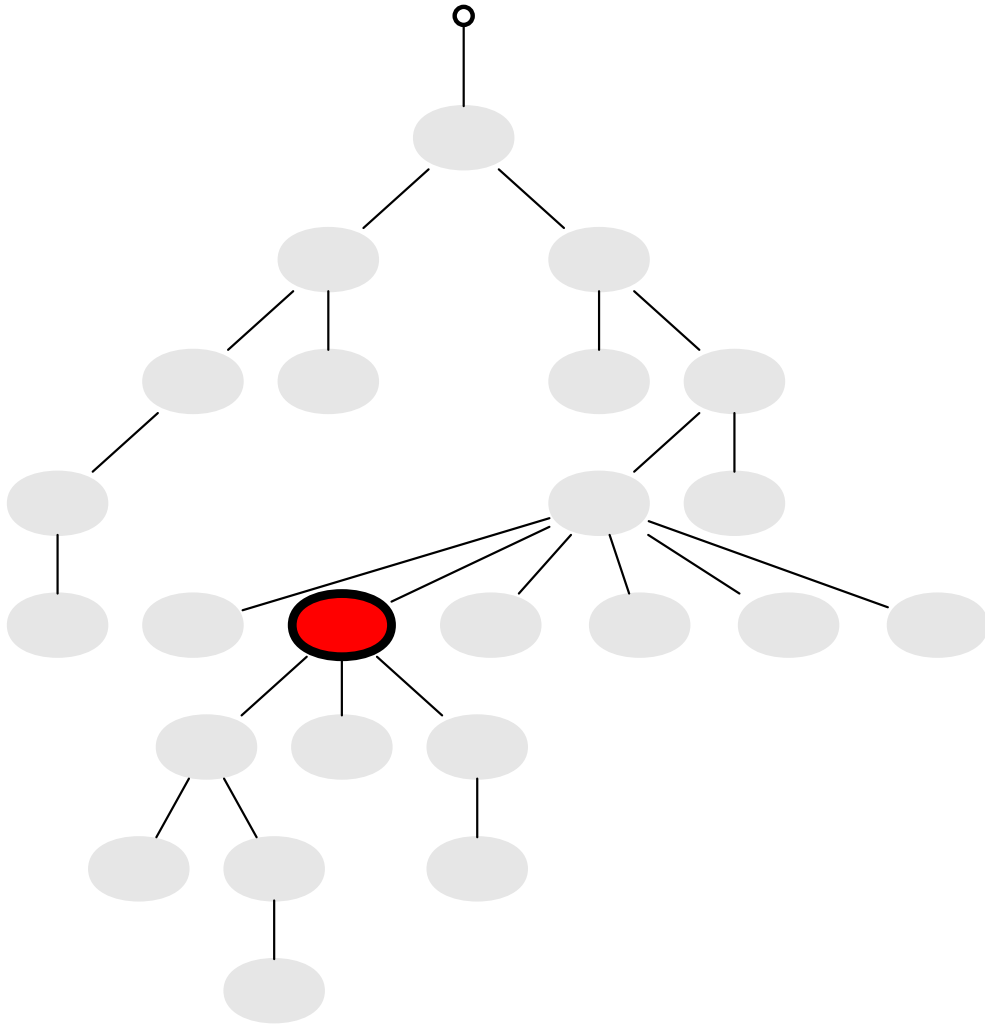
Axes, from a Context Node

`preceding`



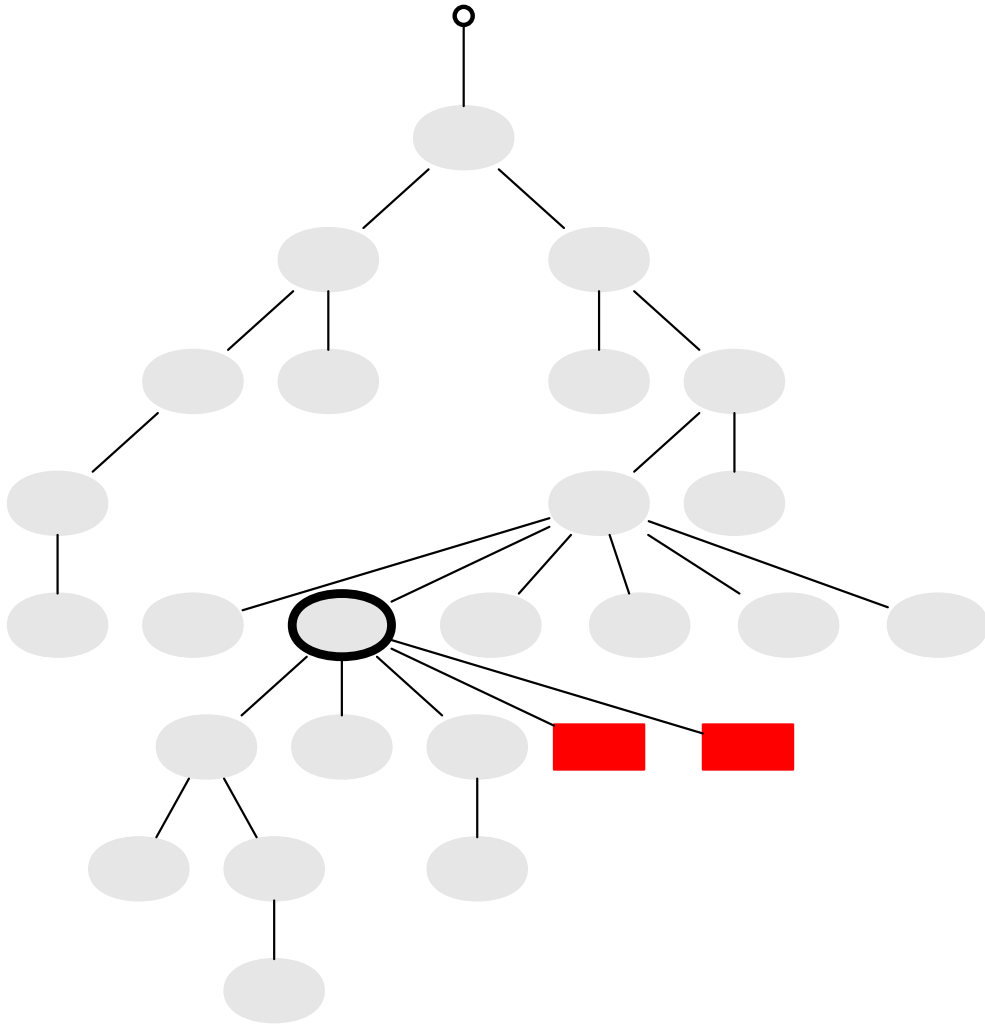
Axes, from a Context Node

`self`



Axes, from a Context Node

`attribute`



Abbreviated Syntax

`/descendant-or-self::node()/` $\rightarrow$ `//`

Abbreviated Syntax

`/descendant-or-self::node()/`	$\longrightarrow$	`//`
`self::node()`	$\longrightarrow$	`.`

Abbreviated Syntax

`/descendant-or-self::node()/`	→	`//`
`self::node()`	→	`.`
`parent::node()`	→	`..`

Abbreviated Syntax

`/descendant-or-self::node()/`	→	`//`
`self::node()`	→	`.`
`parent::node()`	→	`..`
`child::`	→	

Abbreviated Syntax

`/descendant-or-self::node()/`	→	`//`
`self::node()`	→	`.`
`parent::node()`	→	`..`
`child::`	→	
`attribute::`	→	`@`

Abbreviated Syntax

`/descendant-or-self::node()/`	$\longrightarrow$	`//`
`self::node()`	$\longrightarrow$	`.`
`parent::node()`	$\longrightarrow$	`..`
`child::`	$\longrightarrow$	
`attribute::`	$\longrightarrow$	`@`
`position()=`n	$\longrightarrow$	n
	$\vdots$	

Abbreviated Syntax

`/descendant-or-self::node()/`	$\longrightarrow$	`//`
`self::node()`	$\longrightarrow$	`.`
`parent::node()`	$\longrightarrow$	`..`
`child::`	$\longrightarrow$	
`attribute::`	$\longrightarrow$	`@`
`position()=`*n*	$\longrightarrow$	*n*
	$\vdots$	

Thus, for example:

```
/descendant-or-self::node()/child::box/
child::item[attribute::color="yellow"]
```

$\longrightarrow$

Abbreviated Syntax

`/descendant-or-self::node()/`	$\longrightarrow$	`//`
`self::node()`	$\longrightarrow$	`.`
`parent::node()`	$\longrightarrow$	`..`
`child::`	$\longrightarrow$	
`attribute::`	$\longrightarrow$	`@`
`position()=`*n*	$\longrightarrow$	*n*
	$\vdots$	

Thus, for example:

```
/descendant-or-self::node()/child::box/
child::item[attribute::color="yellow"]
```

$\longrightarrow$

```
/descendant-or-self::node()/child::box/
child::item[attribute::color="yellow"]
```

Abbreviated Syntax

`/descendant-or-self::node()/`	$\longrightarrow$	`//`
`self::node()`	$\longrightarrow$	`.`
`parent::node()`	$\longrightarrow$	`..`
`child::`	$\longrightarrow$	
`attribute::`	$\longrightarrow$	`@`
`position()=`*n*	$\longrightarrow$	*n*
	$\vdots$	

Thus, for example:

```
/descendant-or-self::node()/child::box/
child::item[attribute::color="yellow"]
```

$\longrightarrow$

```
//child::box/child::item[attribute::color="yellow"]
```

Abbreviated Syntax

`/descendant-or-self::node()/`	$\longrightarrow$	`//`
`self::node()`	$\longrightarrow$	`.`
`parent::node()`	$\longrightarrow$	`..`
`child::`	$\longrightarrow$	
`attribute::`	$\longrightarrow$	`@`
`position()=`*n*	$\longrightarrow$	*n*
	$\vdots$	

Thus, for example:

```
/descendant-or-self::node()/child::box/
child::item[attribute::color="yellow"]
```

$\longrightarrow$

```
//child::box/child::item[attribute::color="yellow"]
```

Abbreviated Syntax

`/descendant-or-self::node()/`	$\longrightarrow$	`//`
`self::node()`	$\longrightarrow$	`.`
`parent::node()`	$\longrightarrow$	`..`
`child::`	$\longrightarrow$	
`attribute::`	$\longrightarrow$	`@`
`position()=`*n*	$\longrightarrow$	*n*
	$\vdots$	

Thus, for example:

```
/descendant-or-self::node()/child::box/
child::item[attribute::color="yellow"]
```

$\longrightarrow$

```
//box/item[attribute::color="yellow"]
```

Abbreviated Syntax

`/descendant-or-self::node()/`	$\longrightarrow$	`//`
`self::node()`	$\longrightarrow$	`.`
`parent::node()`	$\longrightarrow$	`..`
`child::`	$\longrightarrow$	
`attribute::`	$\longrightarrow$	`@`
`position()=`*n*	$\longrightarrow$	*n*
	$\vdots$	

Thus, for example:

```
/descendant-or-self::node()/child::box/
child::item[attribute::color="yellow"]
```

$\longrightarrow$

```
//box/item[attribute::color="yellow"]
```

Abbreviated Syntax

`/descendant-or-self::node()/`	$\longrightarrow$	`//`
`self::node()`	$\longrightarrow$	`.`
`parent::node()`	$\longrightarrow$	`..`
`child::`	$\longrightarrow$	
`attribute::`	$\longrightarrow$	`@`
`position()=`*n*	$\longrightarrow$	*n*
	$\vdots$	

Thus, for example:

```
/descendant-or-self::node()/child::box/
child::item[attribute::color="yellow"]
```

$\longrightarrow$

```
//box/item[@color="yellow"]
```

Abbreviated Syntax

`/descendant-or-self::node()/`	$\rightarrow$	`//`
`self::node()`	$\rightarrow$	`.`
`parent::node()`	$\rightarrow$	`..`
`child::`	$\rightarrow$	
`attribute::`	$\rightarrow$	`@`
`position()=`*n*	$\rightarrow$	*n*
	$\vdots$	

Thus, for example:

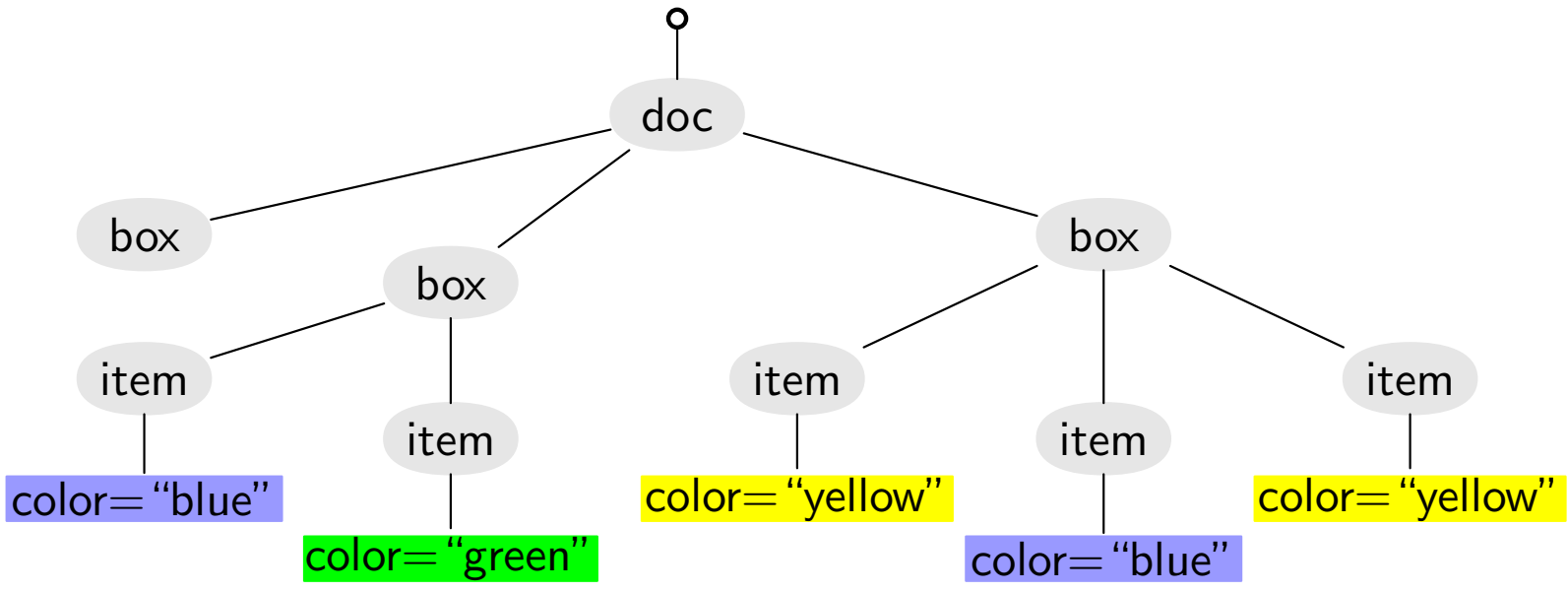
```
/descendant-or-self::node()/child::box/
child::item[attribute::color="yellow"]
```

$\rightarrow$

```
//box/item[@color="yellow"]
```

Another Example

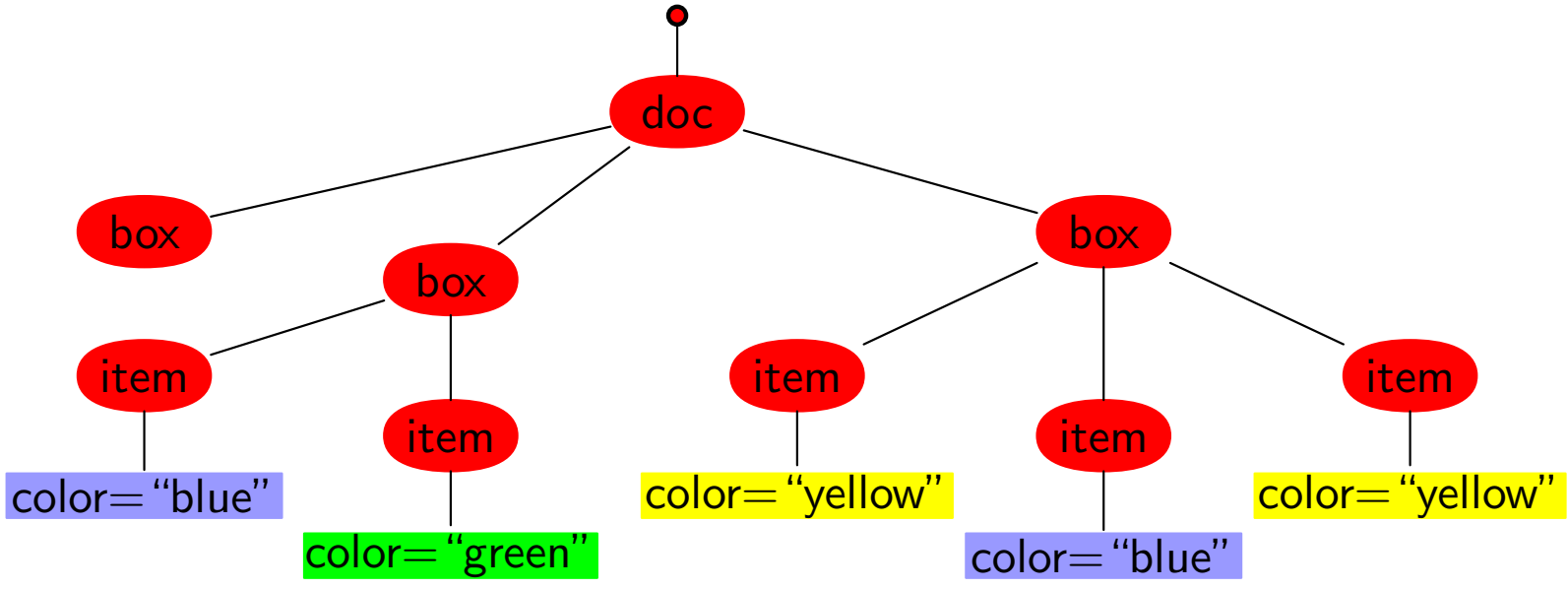
Document:



Query: `//box/item[@color="yellow"]`

Another Example

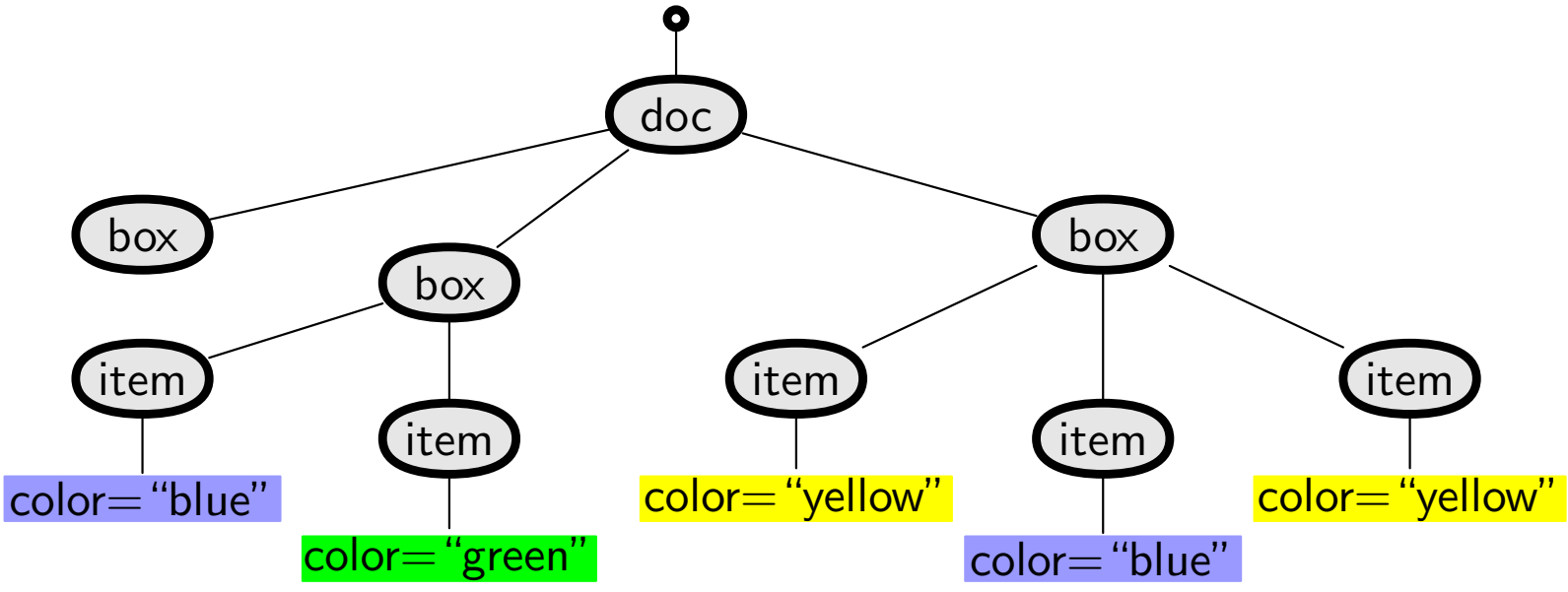
Document:



Query: `//box/item[@color="yellow"]`

Another Example

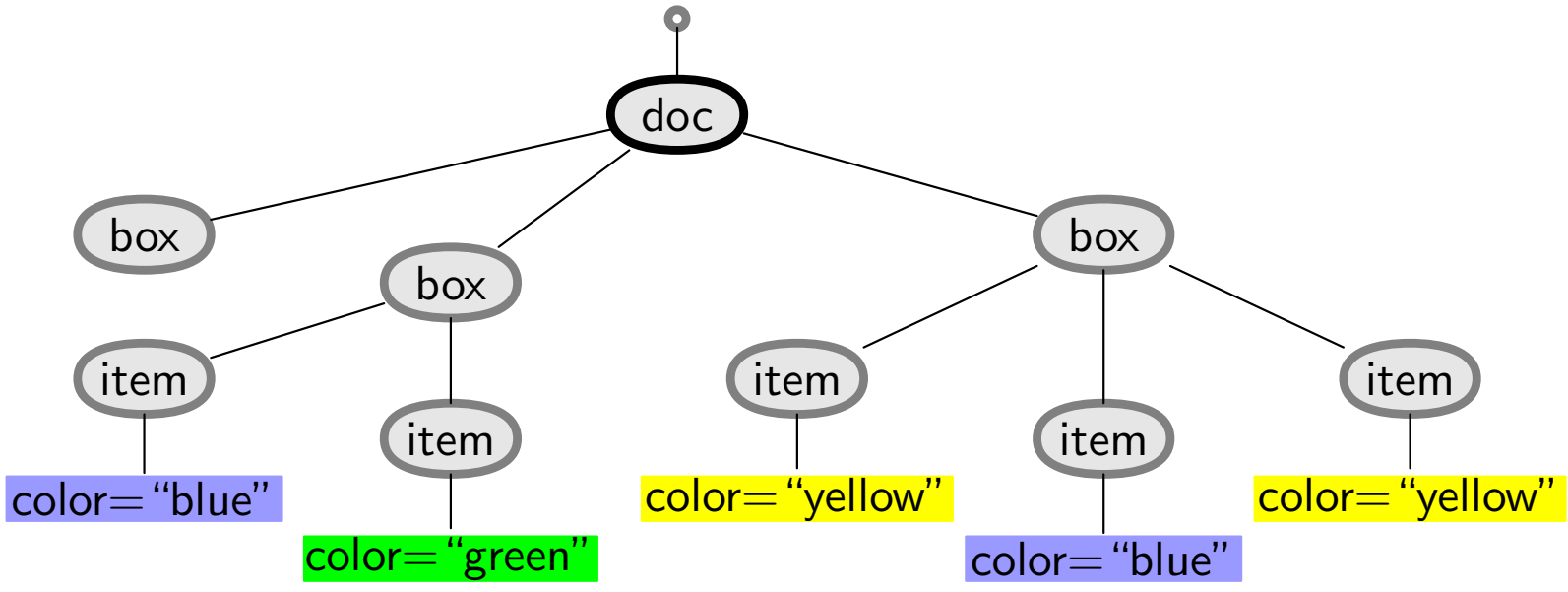
Document:



Query: `//box/item[@color="yellow"]`

Another Example

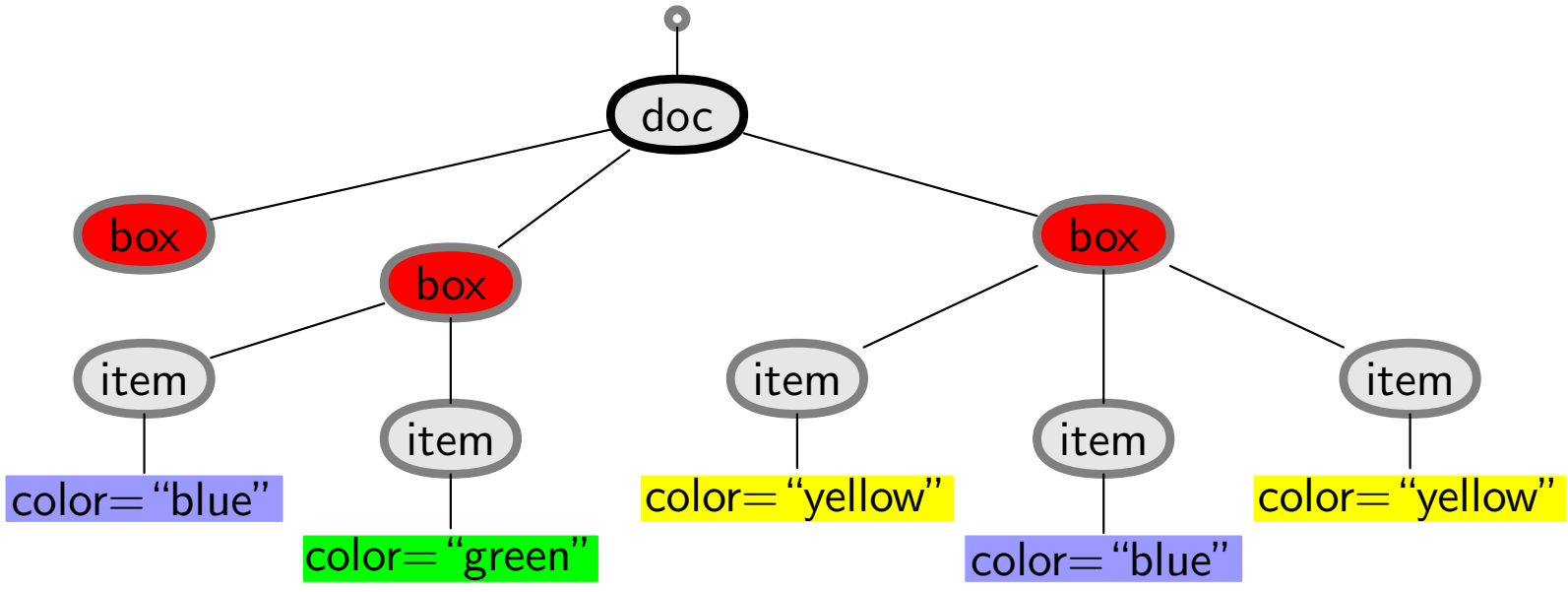
Document:



Query: `//box/item[@color="yellow"]`

Another Example

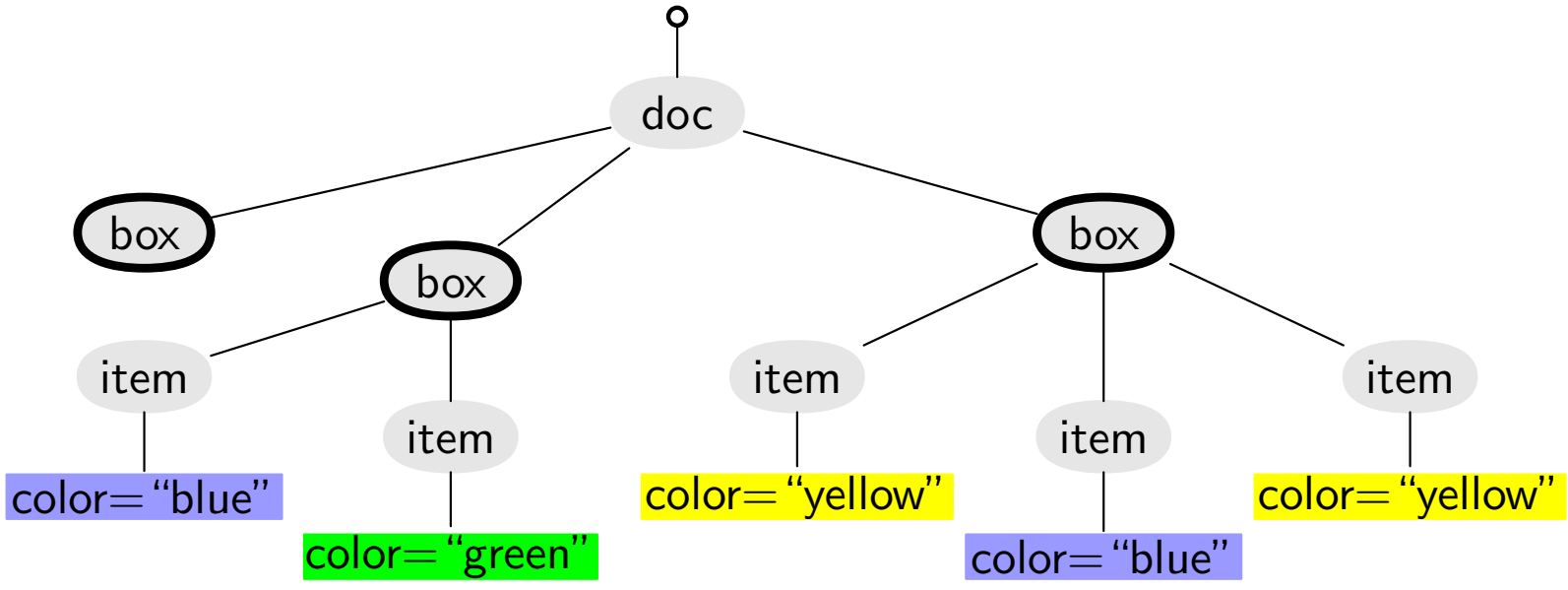
Document:



Query: `//box/item[@color="yellow"]`

Another Example

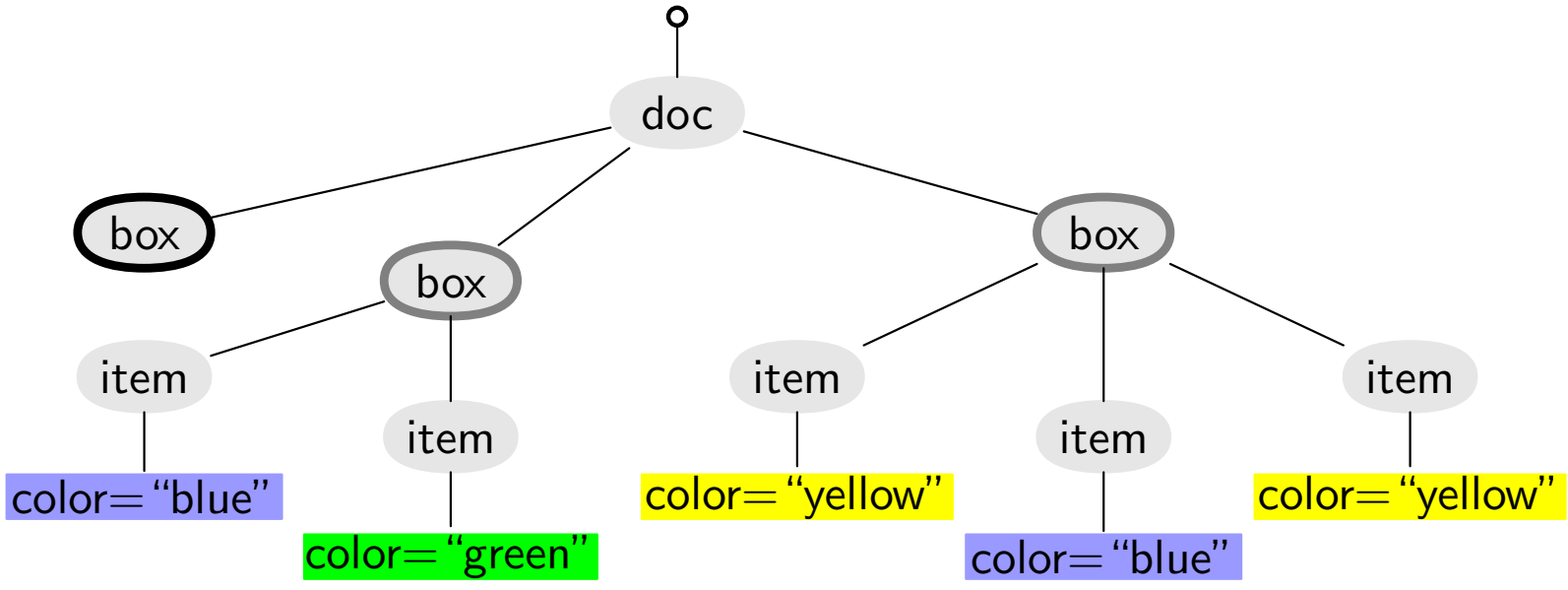
Document:



Query: `//box/item[@color="yellow"]`

Another Example

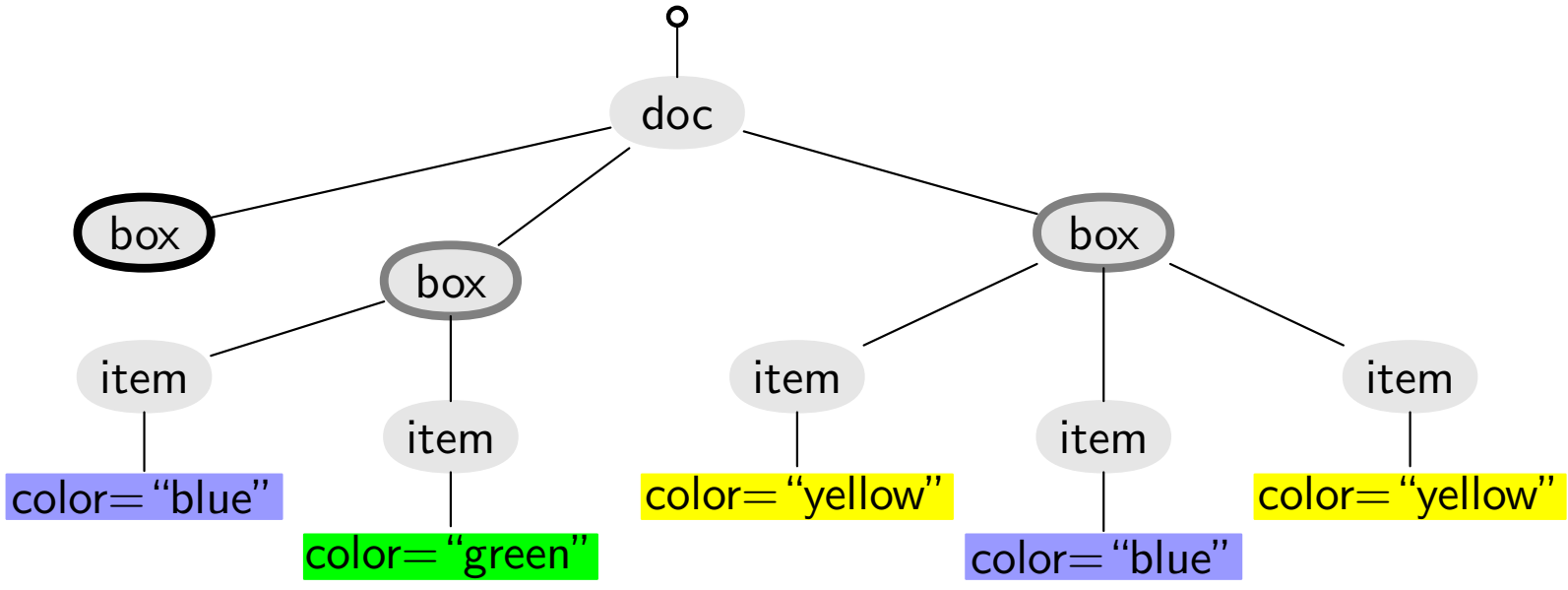
Document:



Query: `//box/item[@color="yellow"]`

Another Example

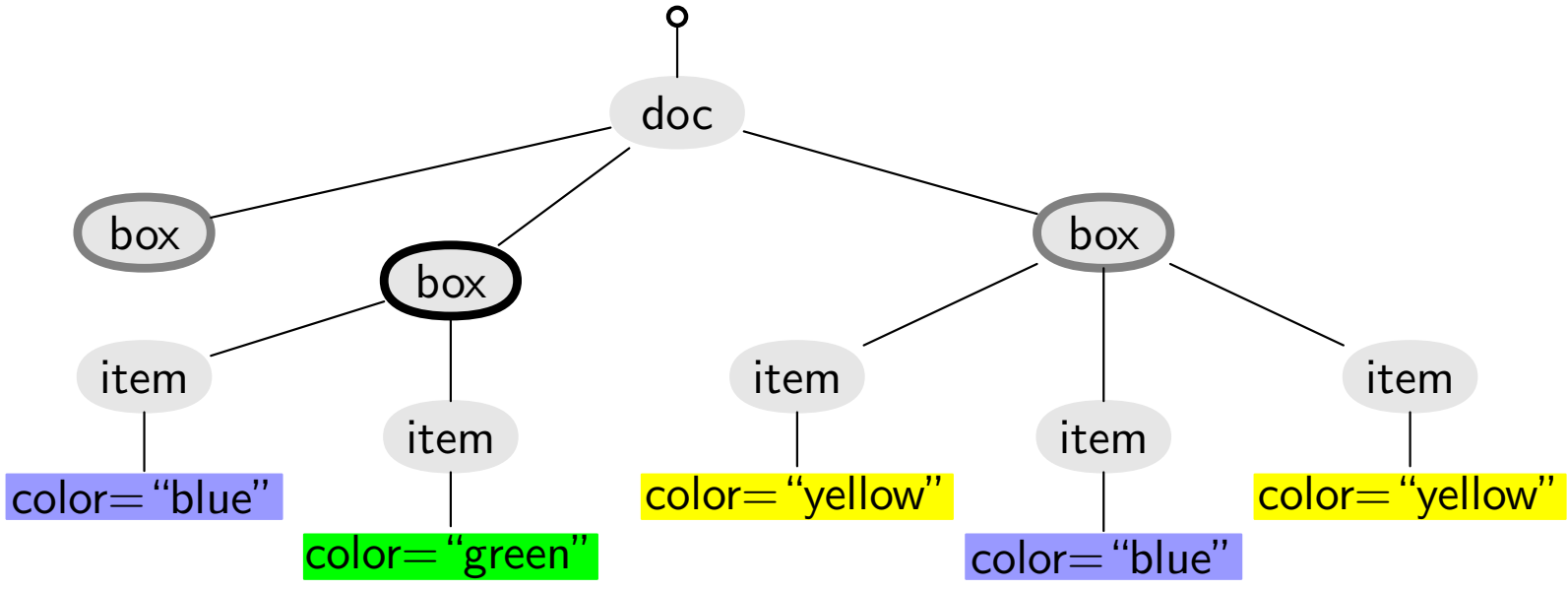
Document:



Query: `//box/item[@color="yellow"]`

Another Example

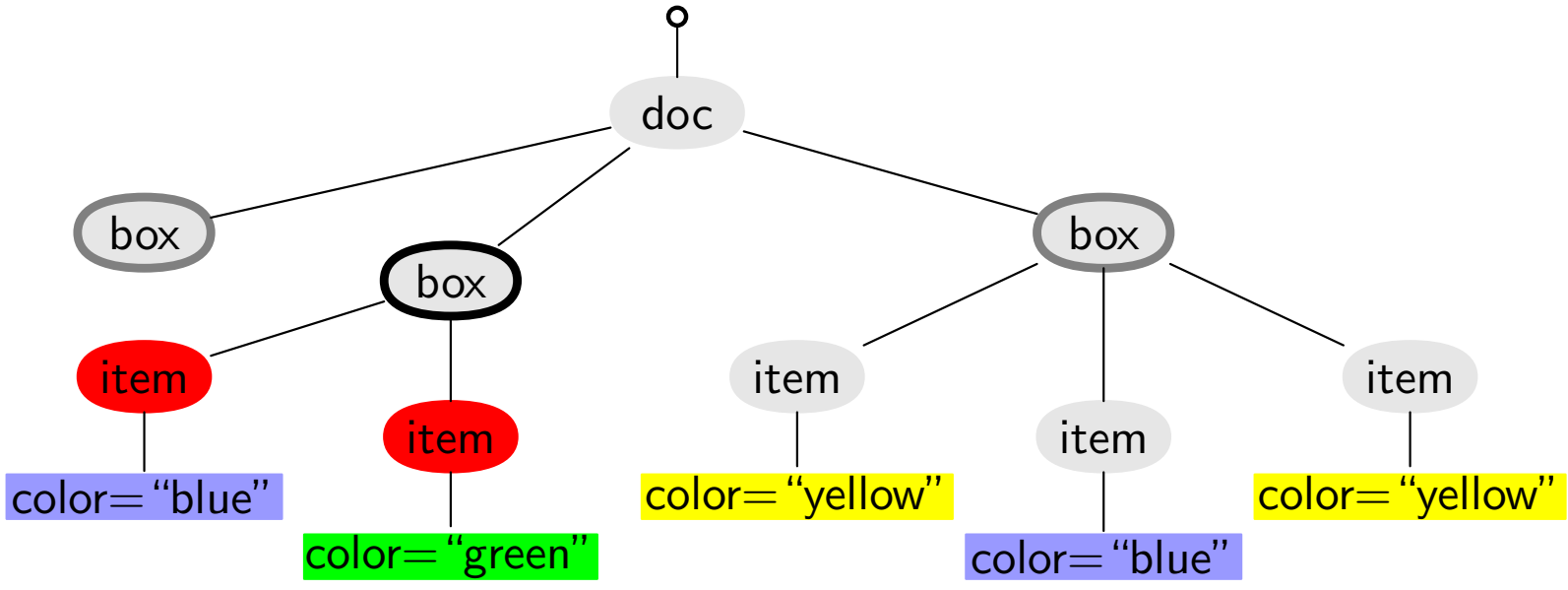
Document:



Query: `//box/item[@color="yellow"]`

Another Example

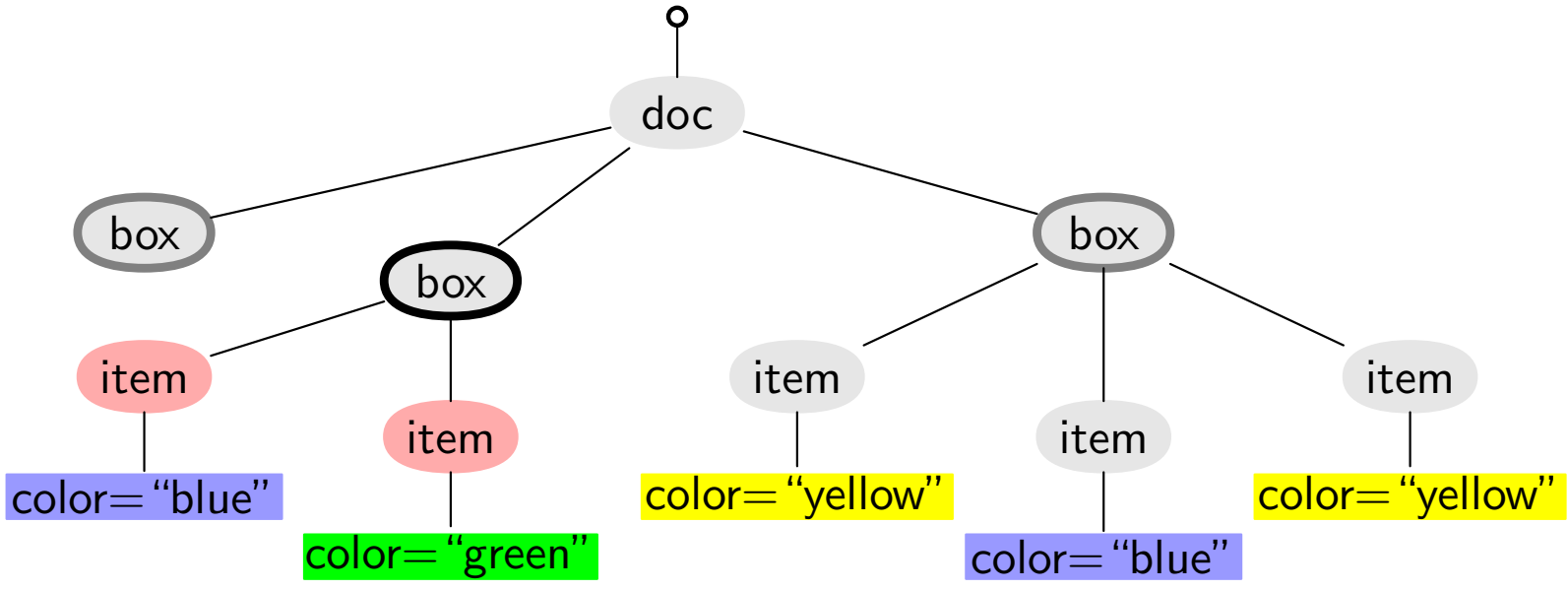
Document:



Query: `//box/item[@color="yellow"]`

Another Example

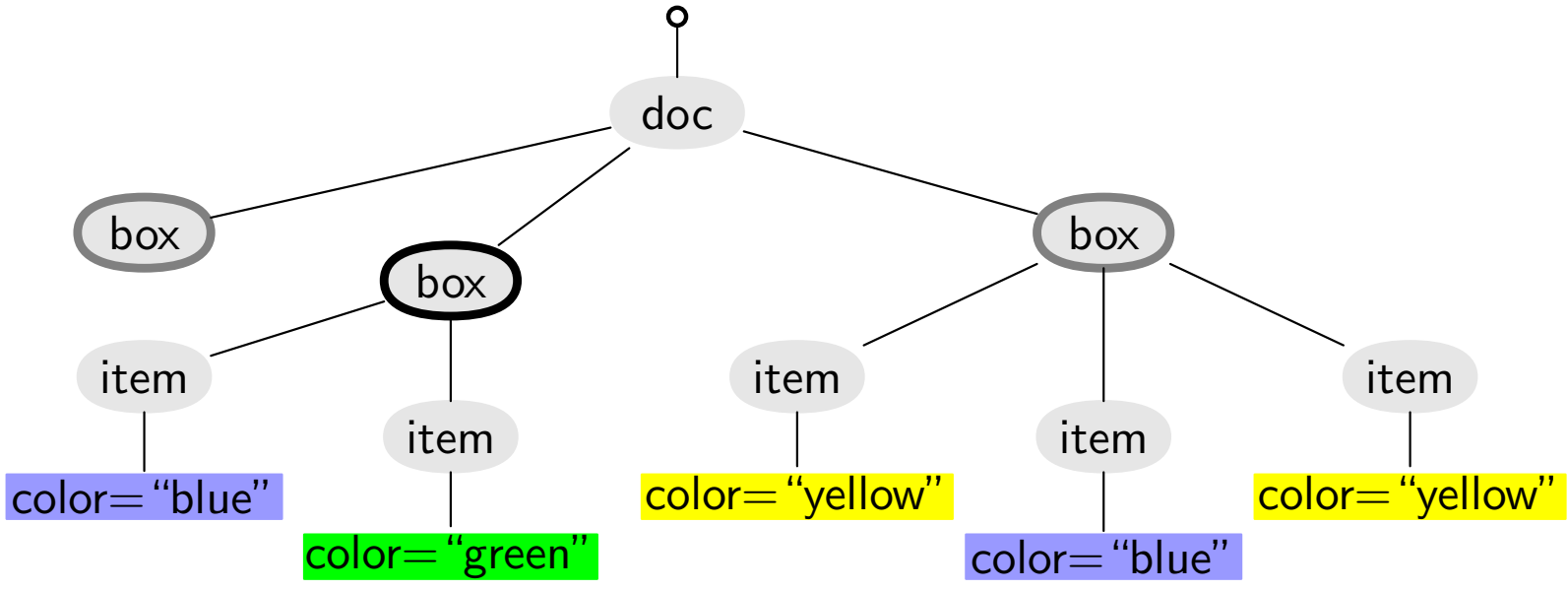
Document:



Query: `//box/item[@color="yellow"]`

Another Example

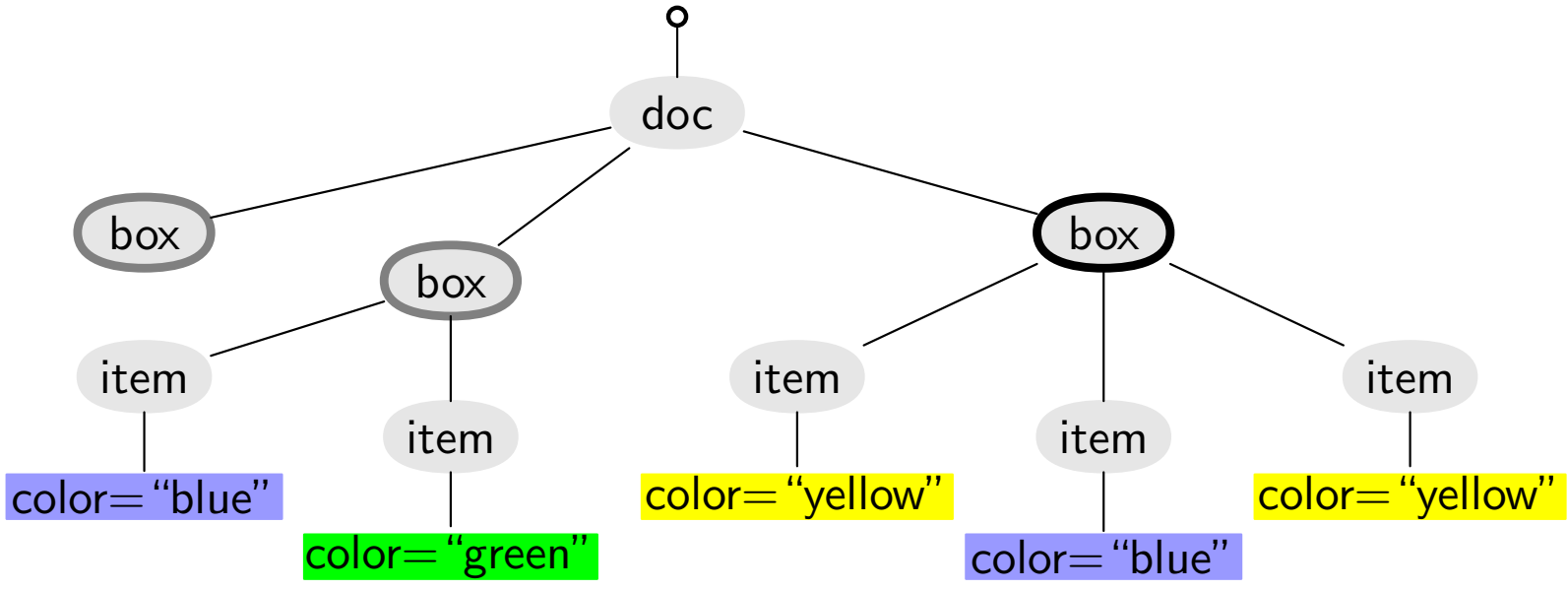
Document:



Query: `//box/item[@color="yellow"]`

Another Example

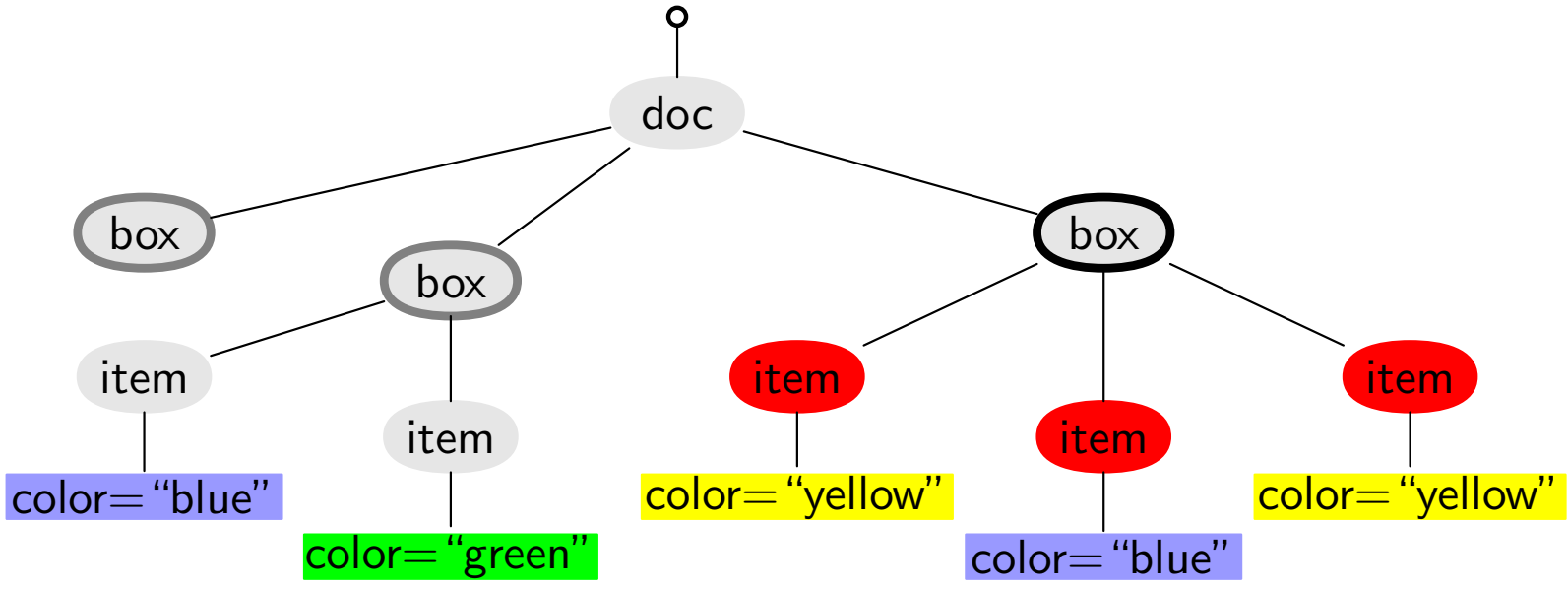
Document:



Query: `//box/item[@color="yellow"]`

Another Example

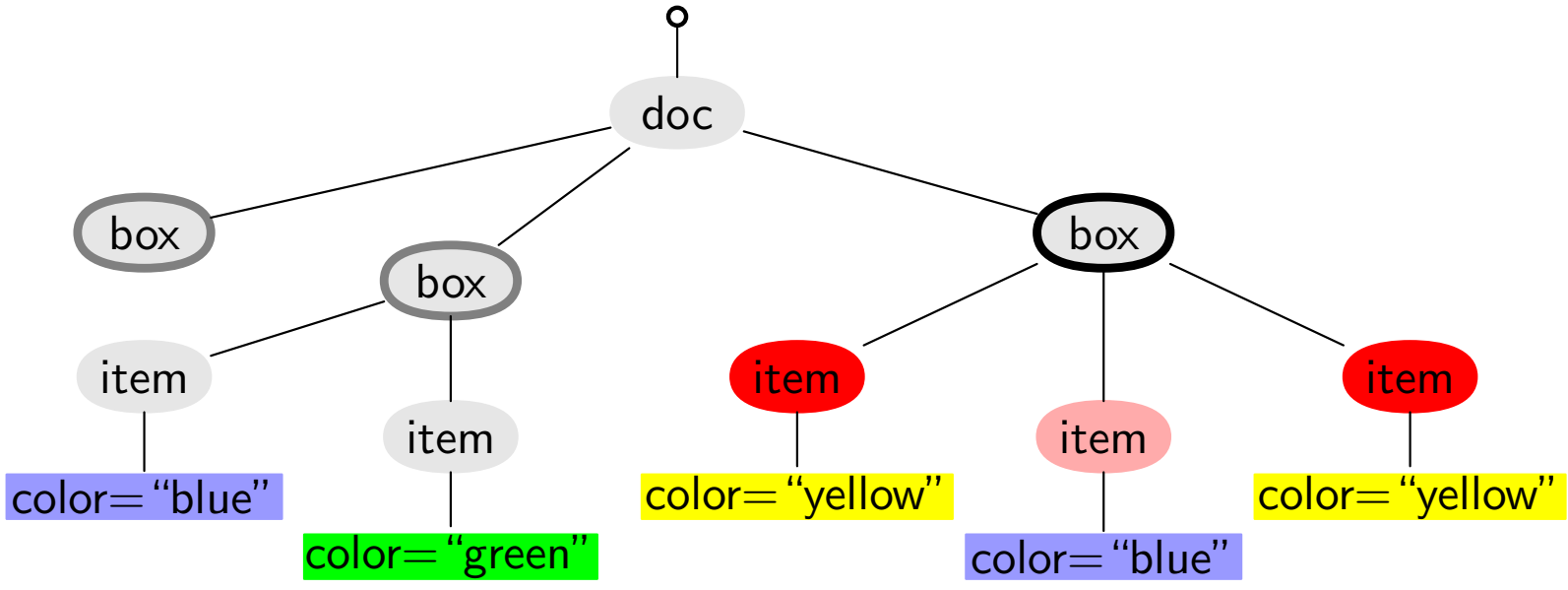
Document:



Query: `//box/item[@color="yellow"]`

Another Example

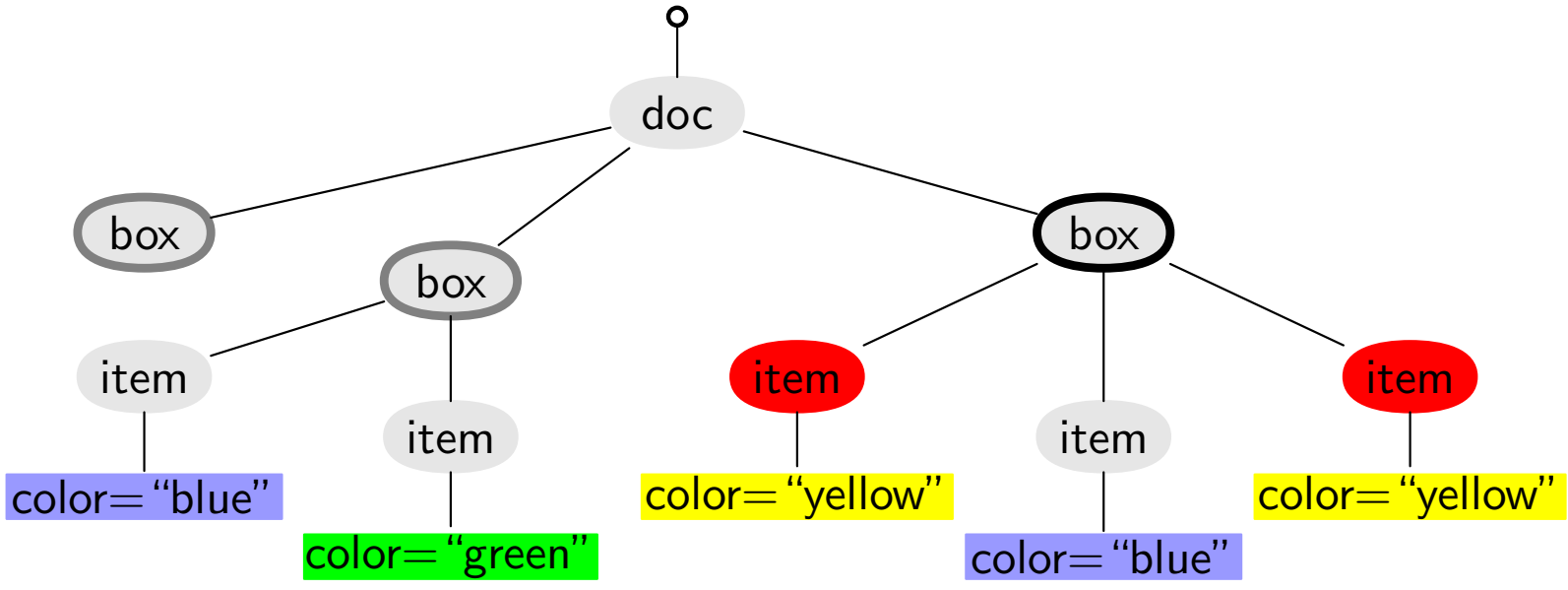
Document:



Query: `//box/item[@color="yellow"]`

Another Example

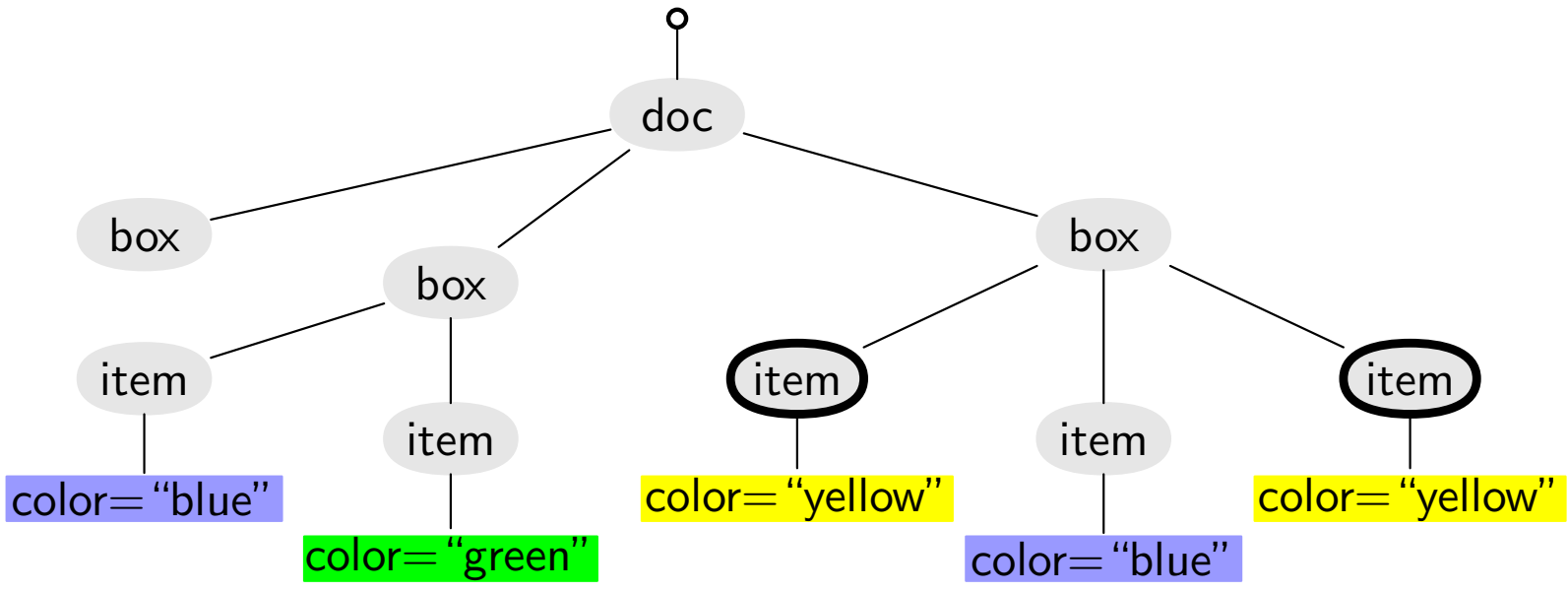
Document:



Query: `//box/item[@color="yellow"]`

Another Example

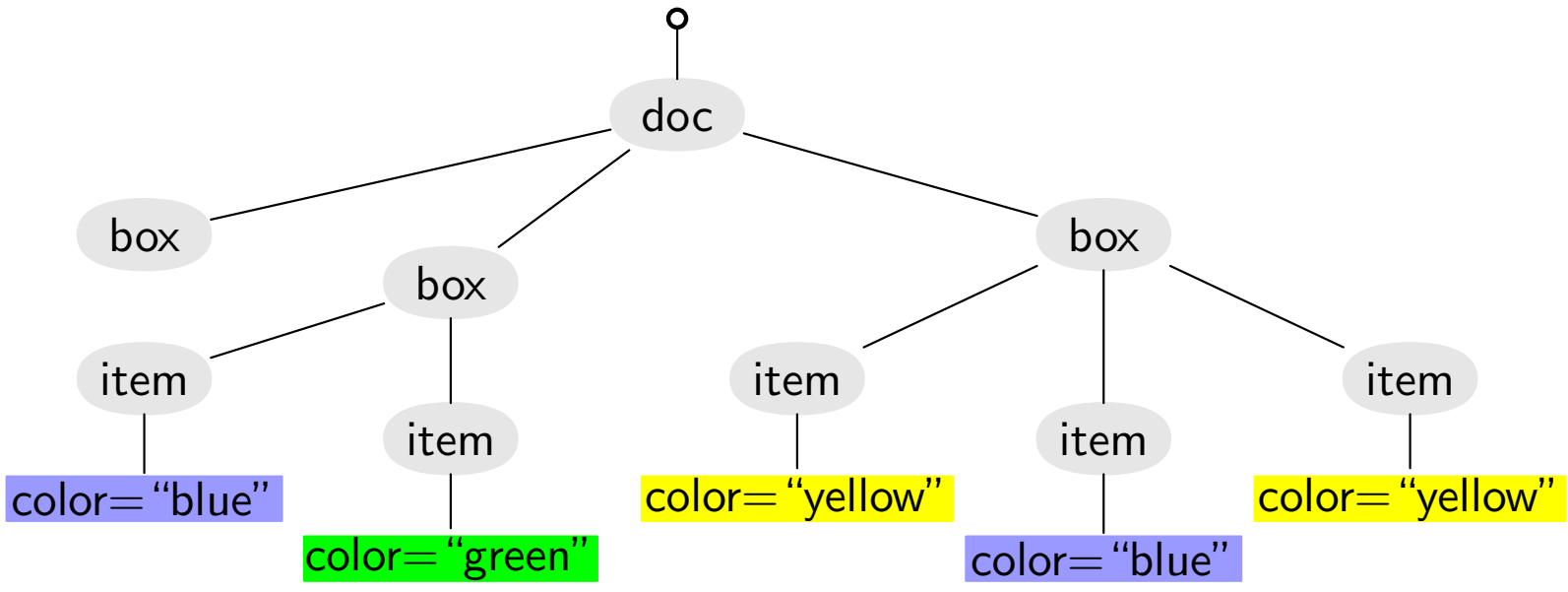
Document:



Query: `//box/item[@color="yellow"]`

Paths as Filter Expressions

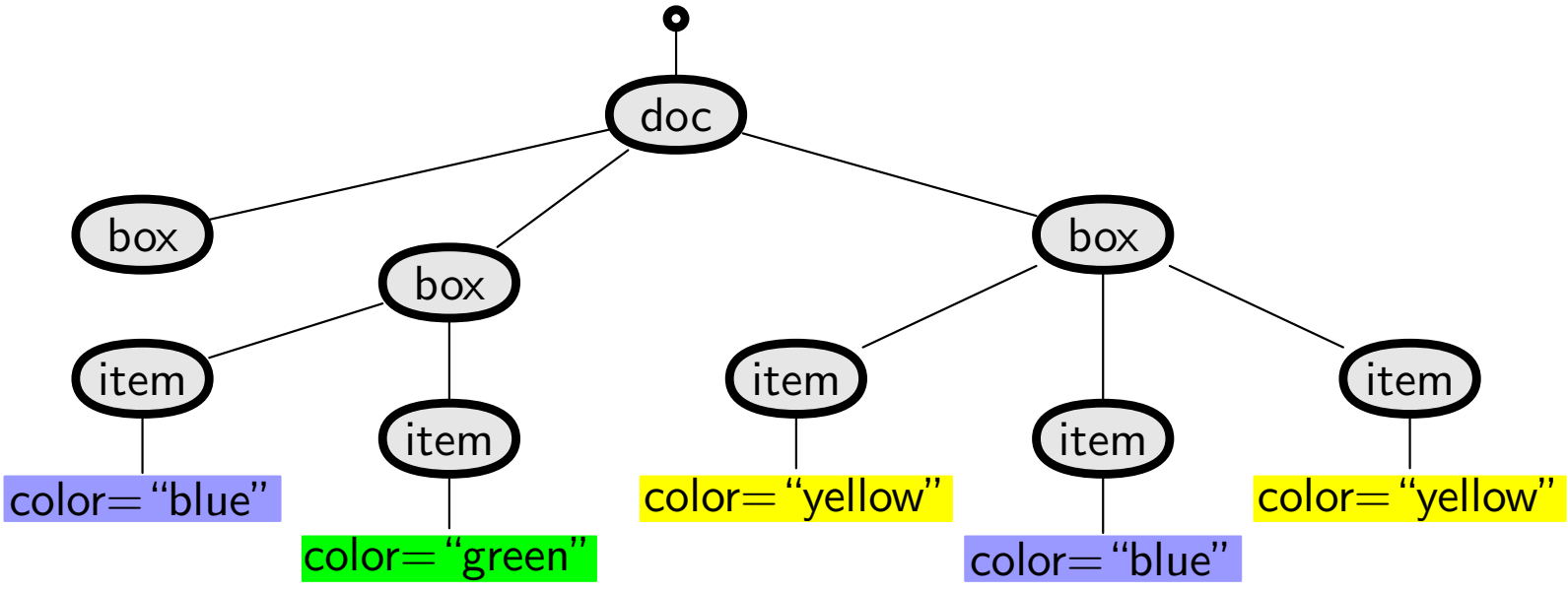
Document:



Query: `//box[item]`

Paths as Filter Expressions

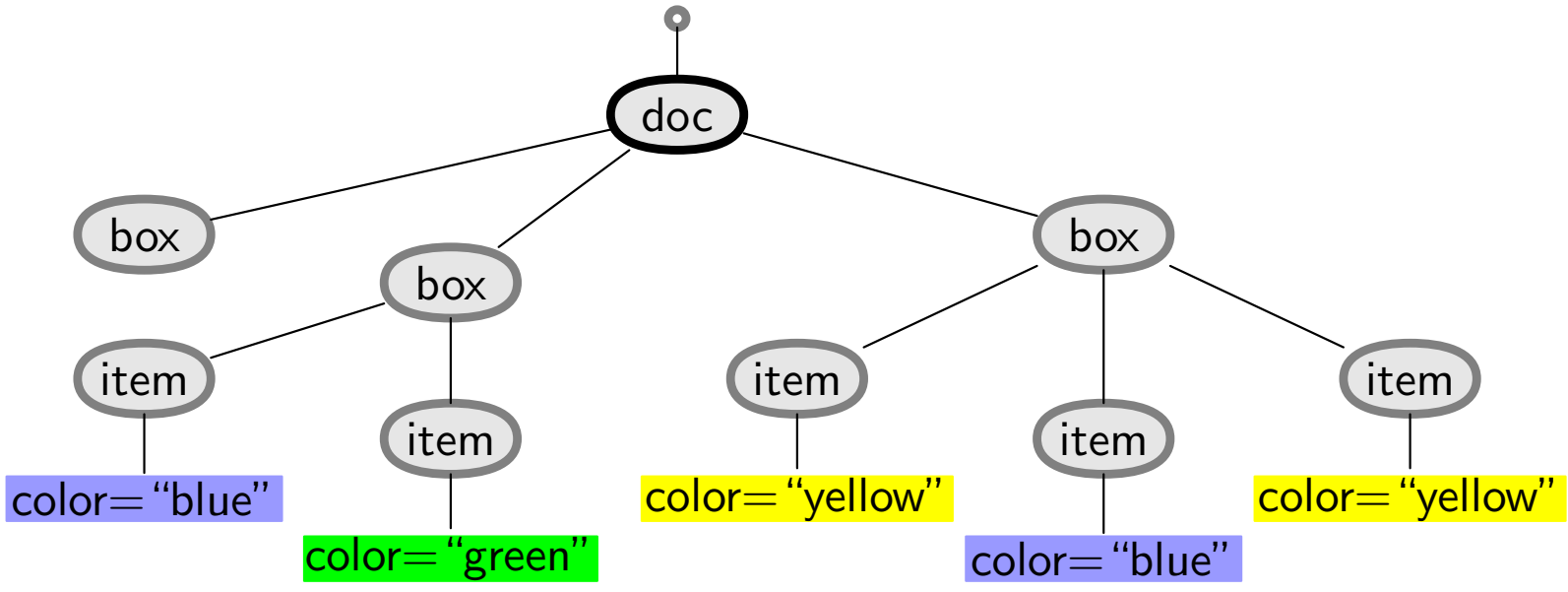
Document:



Query: `//box[item]`

Paths as Filter Expressions

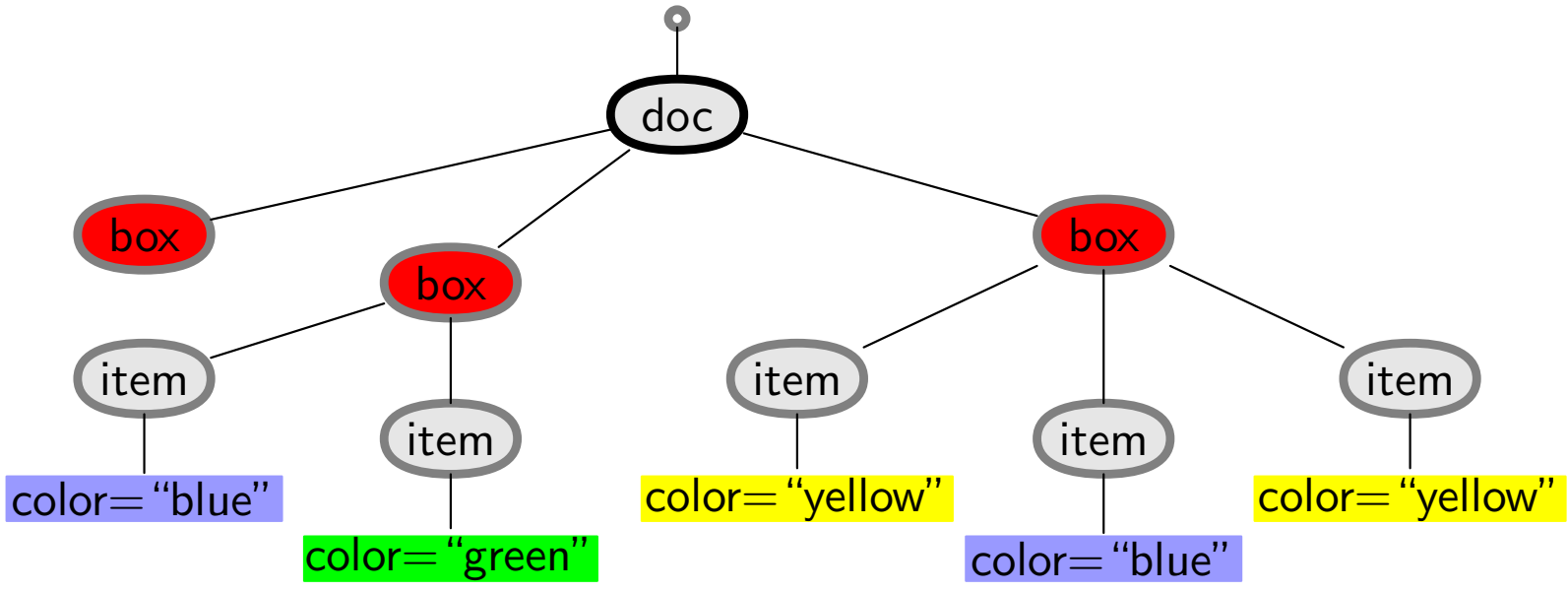
Document:



Query: `//box[item]`

Paths as Filter Expressions

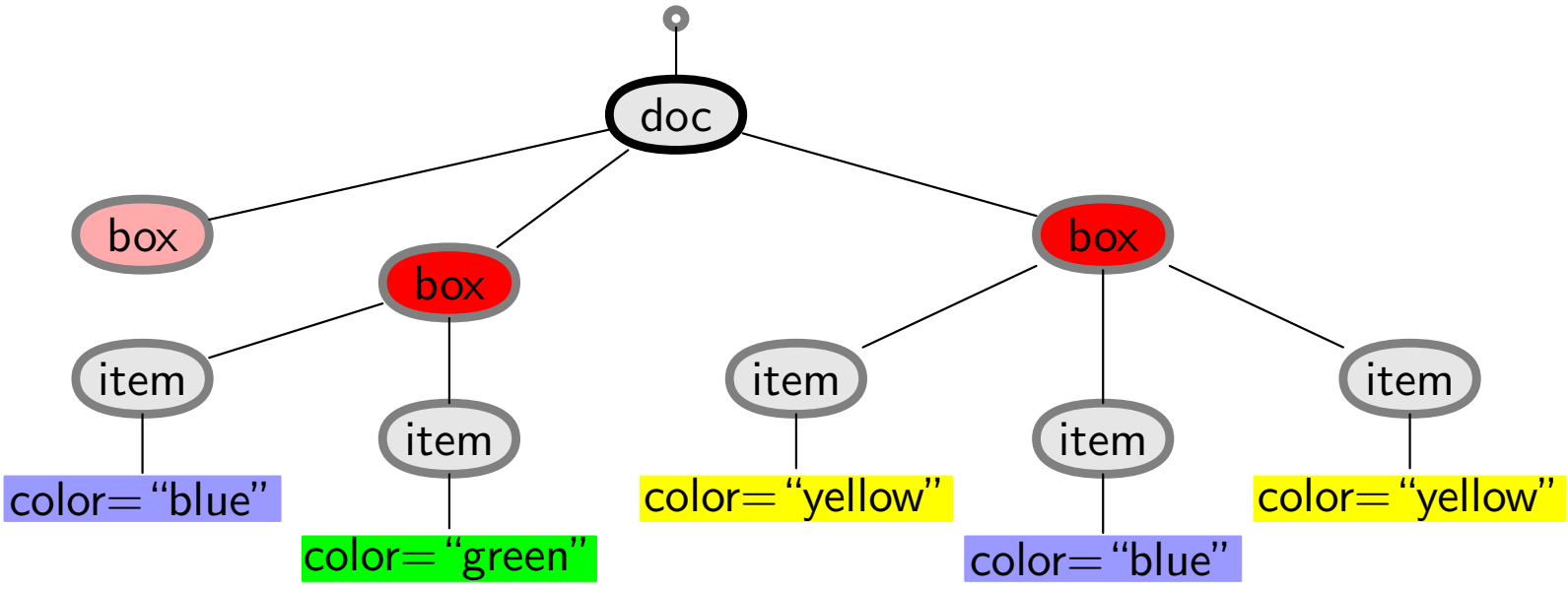
Document:



Query: `//box[item]`

Paths as Filter Expressions

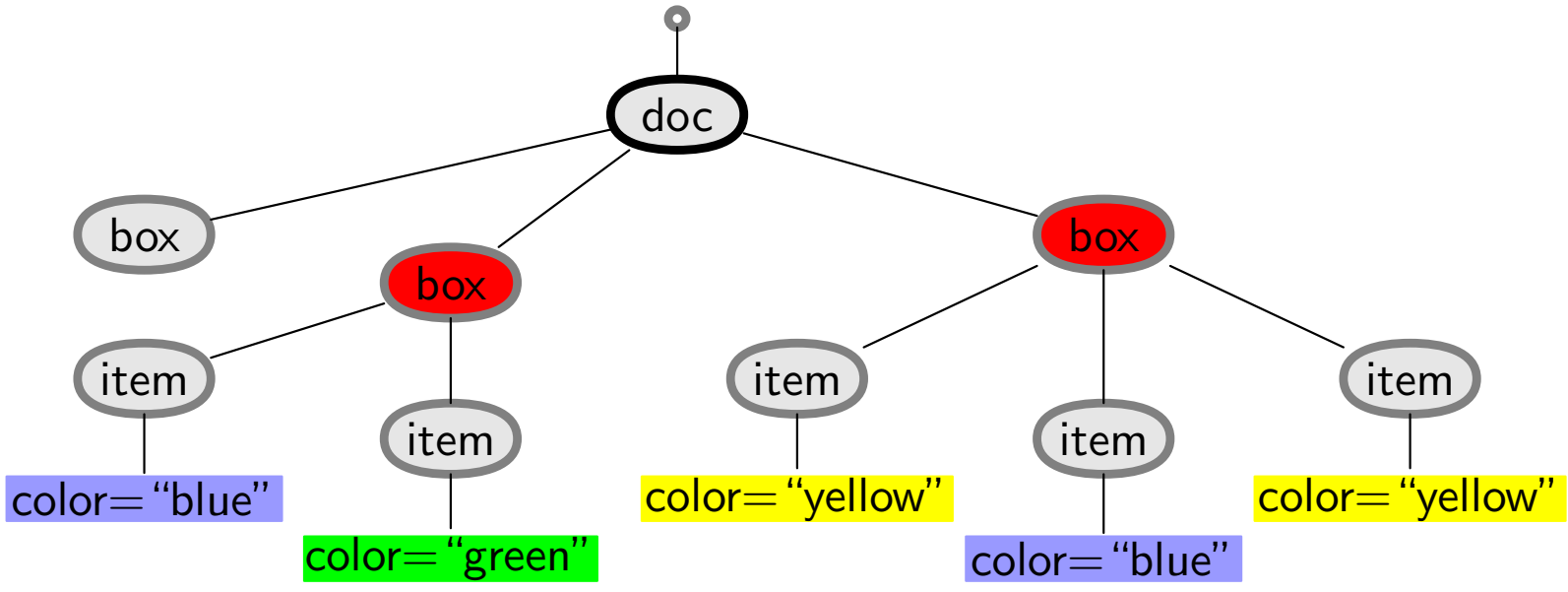
Document:



Query: `//box[item]`

Paths as Filter Expressions

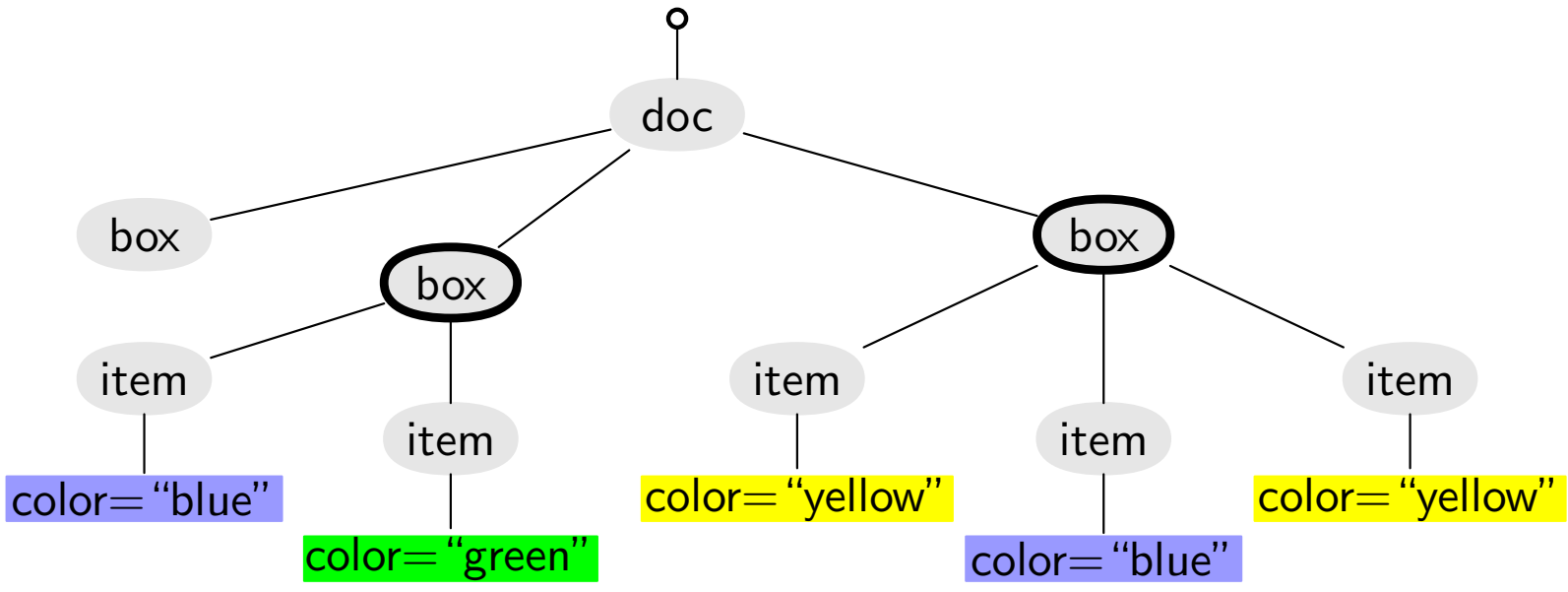
Document:



Query: `//box[item]`

Paths as Filter Expressions

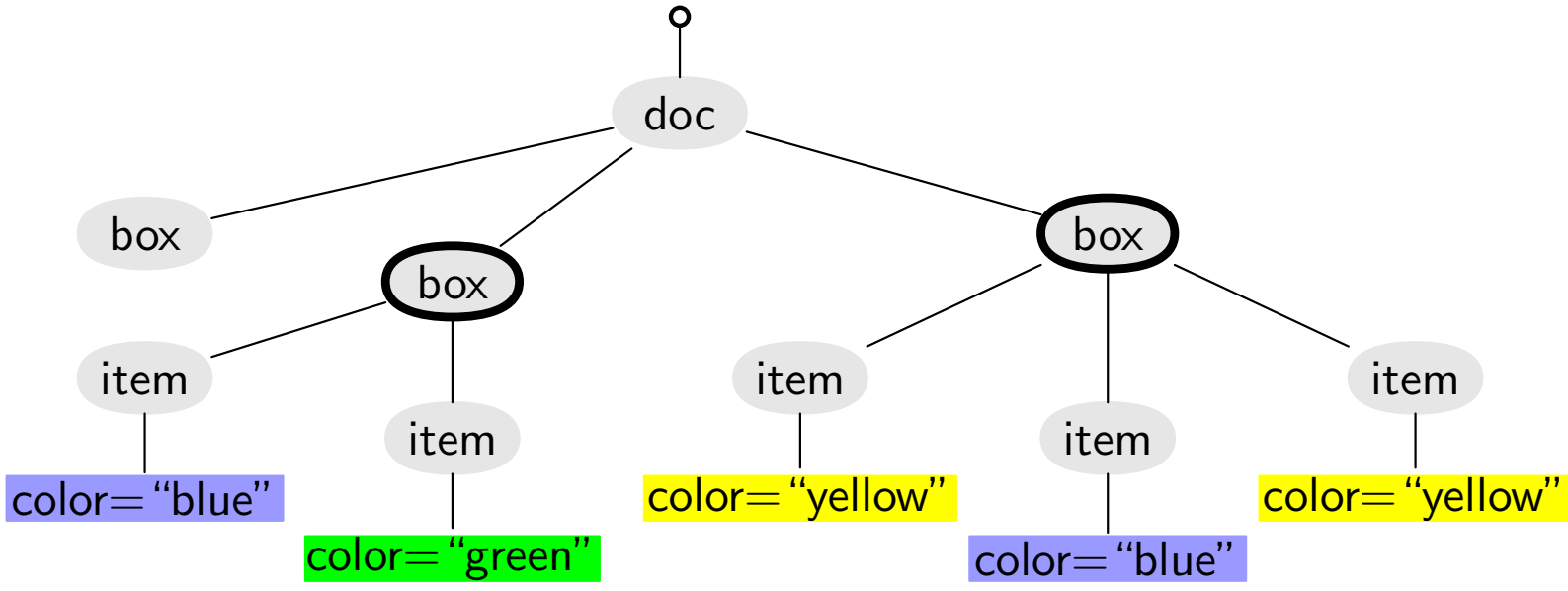
Document:



Query: `//box[item]`

Paths as Filter Expressions

Document:

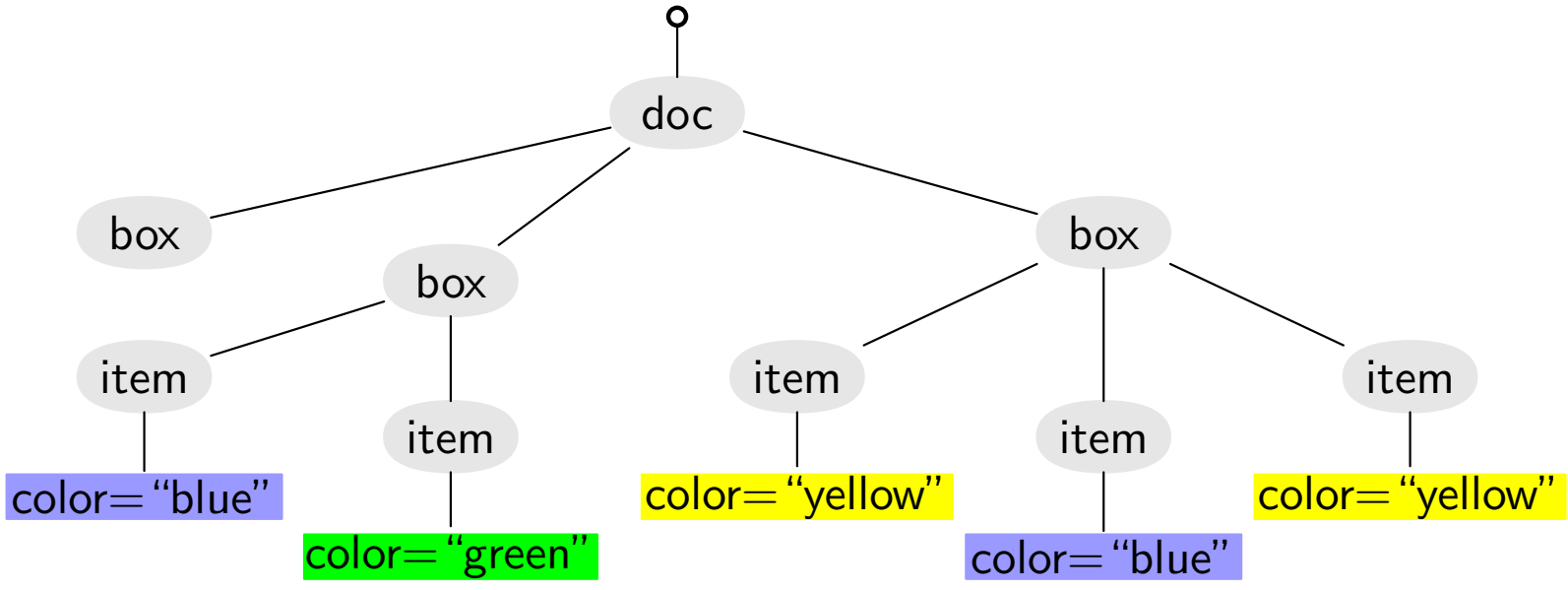


Query: `//box[item]`

Note the difference to `//box/item` !

Nested Filter Expressions

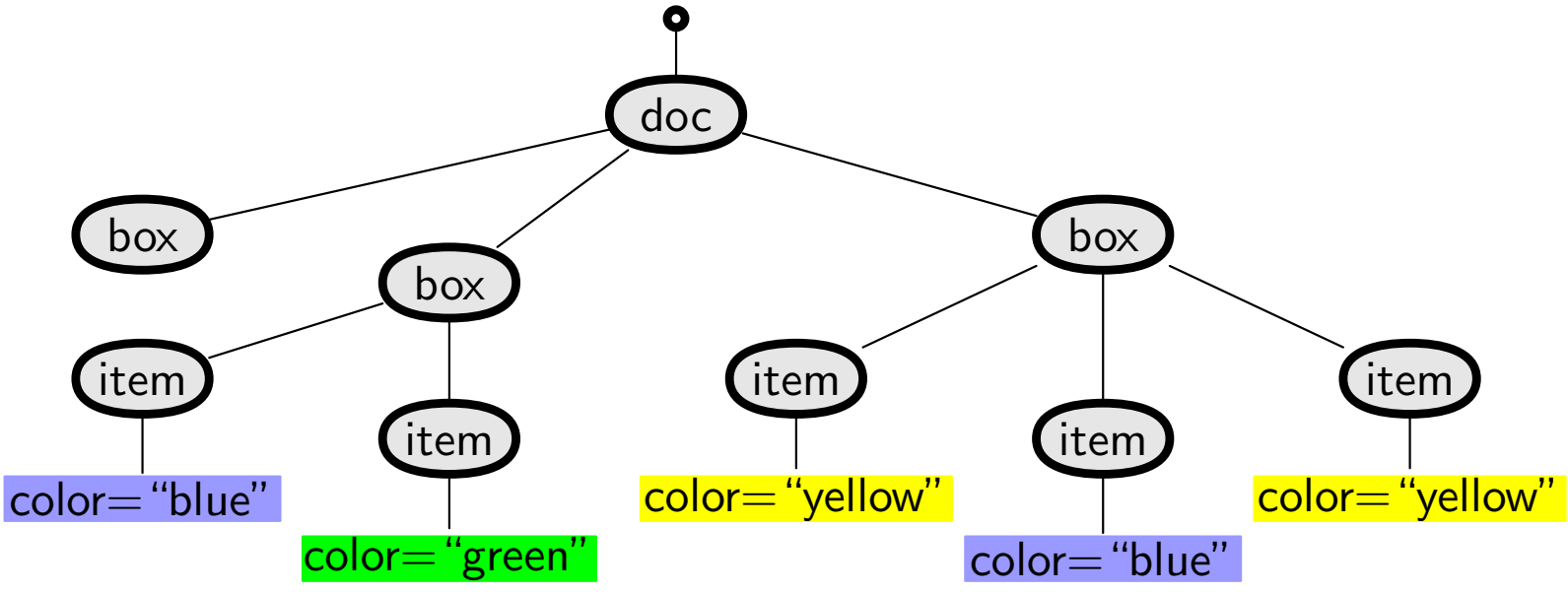
Document:



Query: `//box[item[@color="yellow"]]`

Nested Filter Expressions

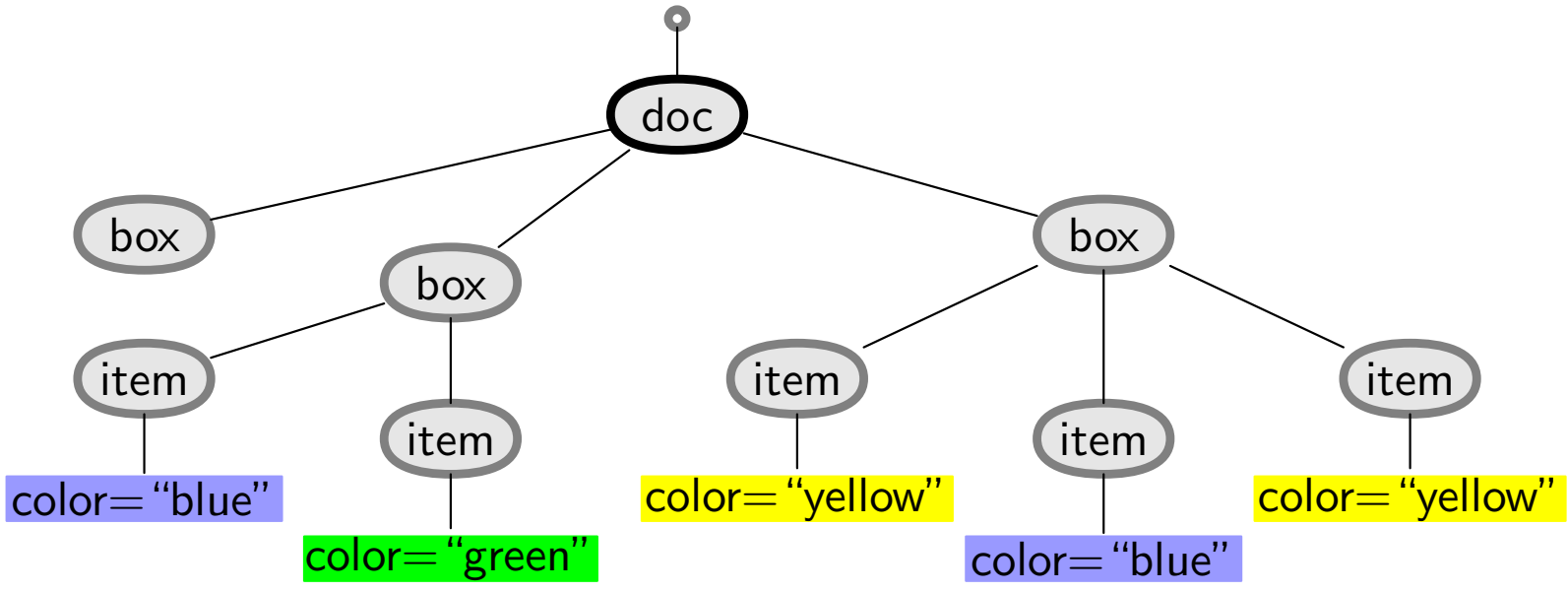
Document:



Query: `//box[item[@color="yellow"]]`

Nested Filter Expressions

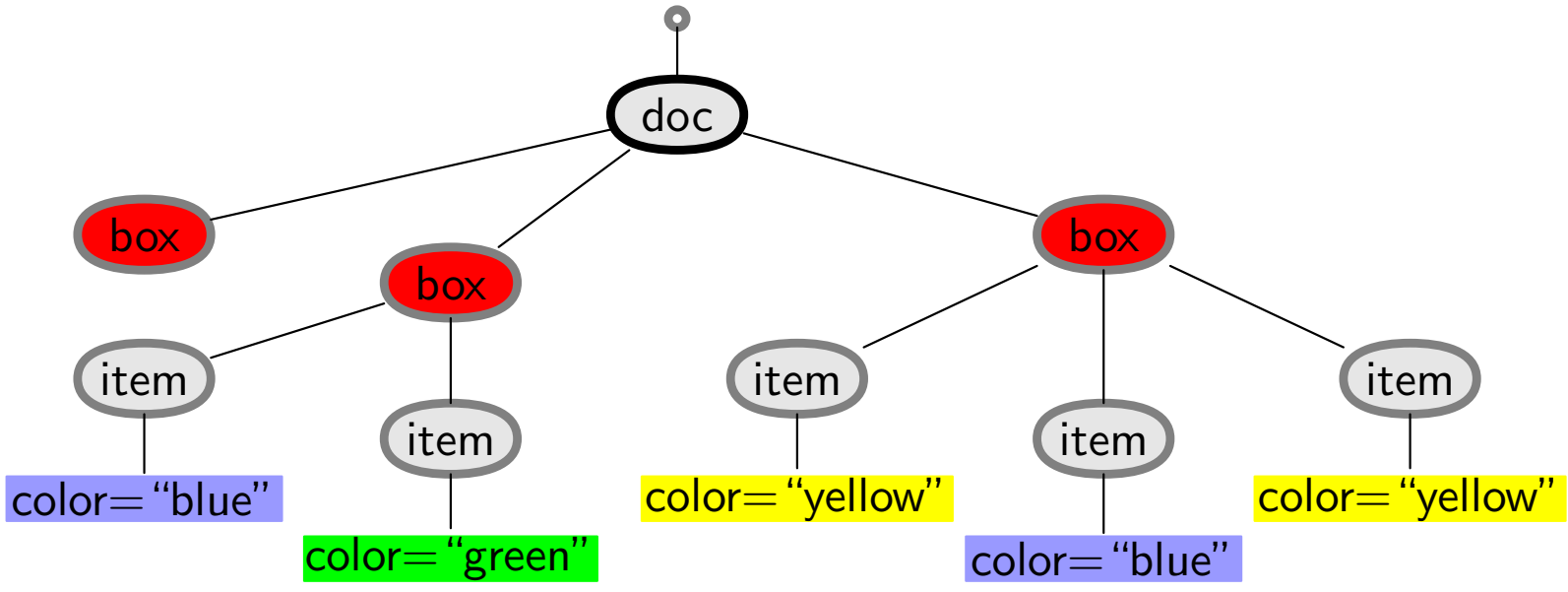
Document:



Query: `//box[item[@color="yellow"]]`

Nested Filter Expressions

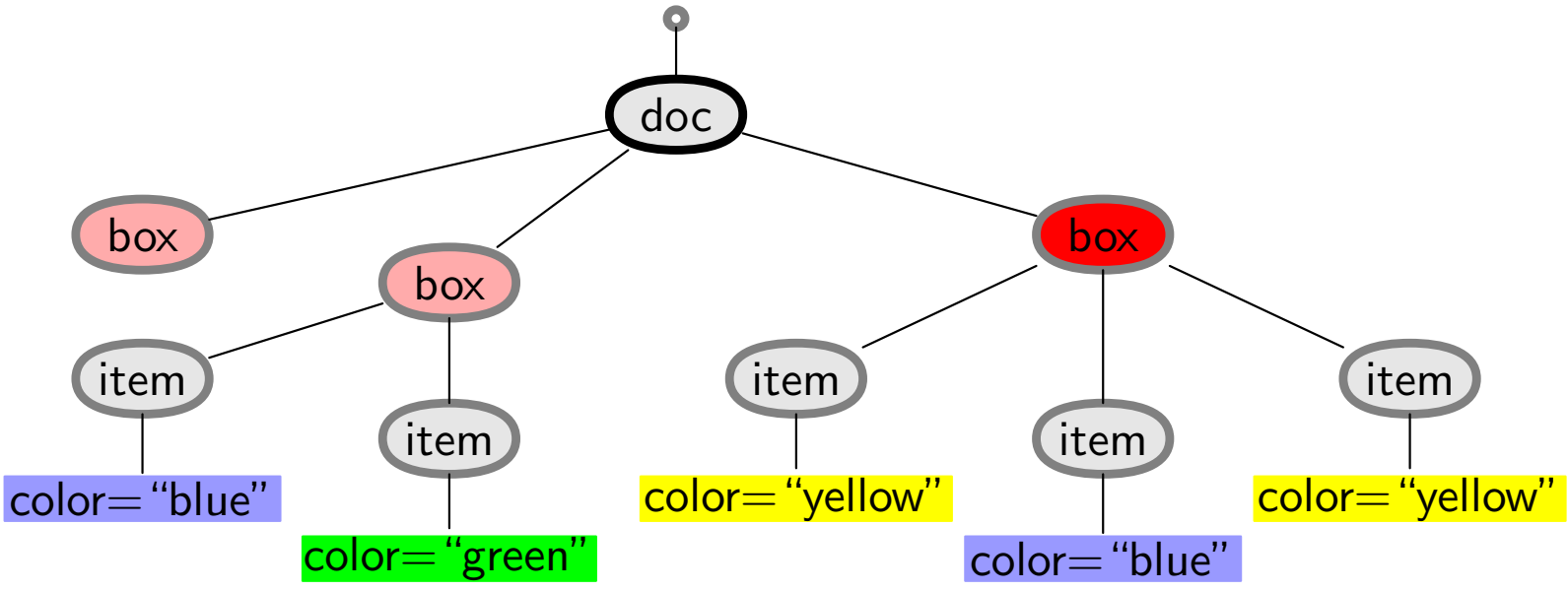
Document:



Query: `//box[item[@color="yellow"]]`

Nested Filter Expressions

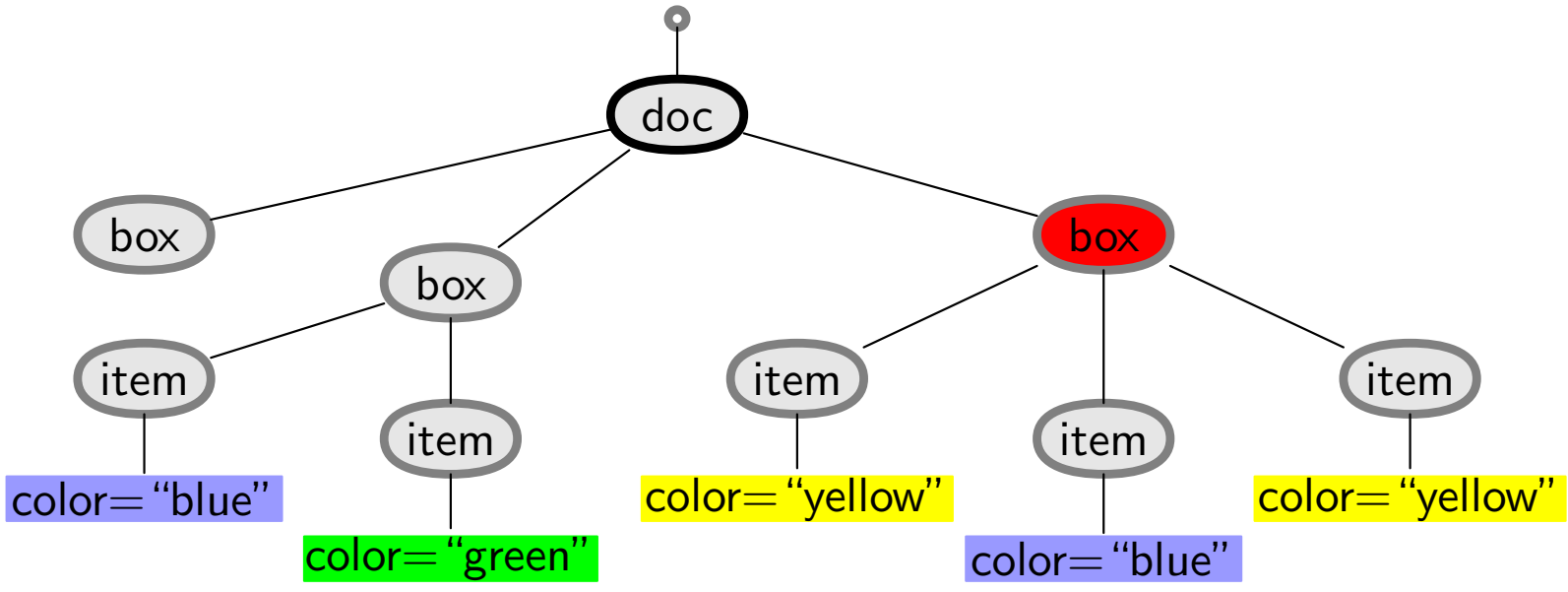
Document:



Query: `//box[item[@color="yellow"]]`

Nested Filter Expressions

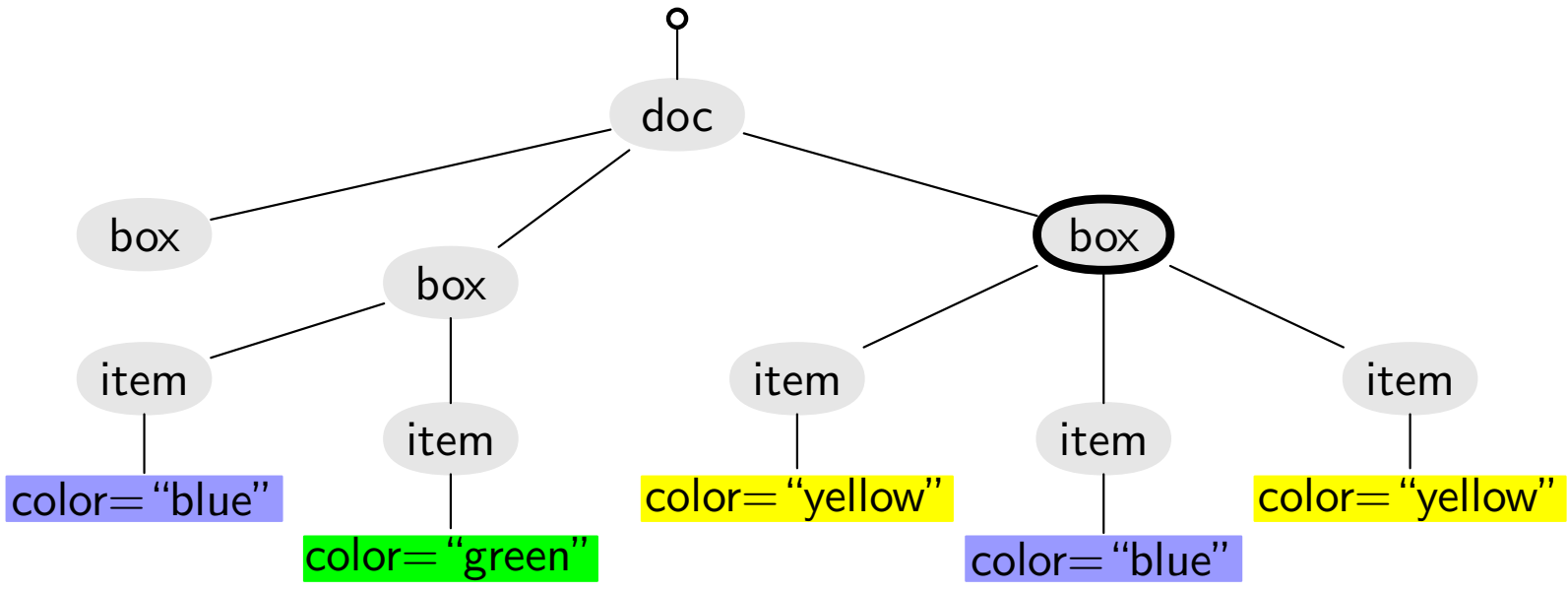
Document:



Query: `//box[item[@color="yellow"]]`

Nested Filter Expressions

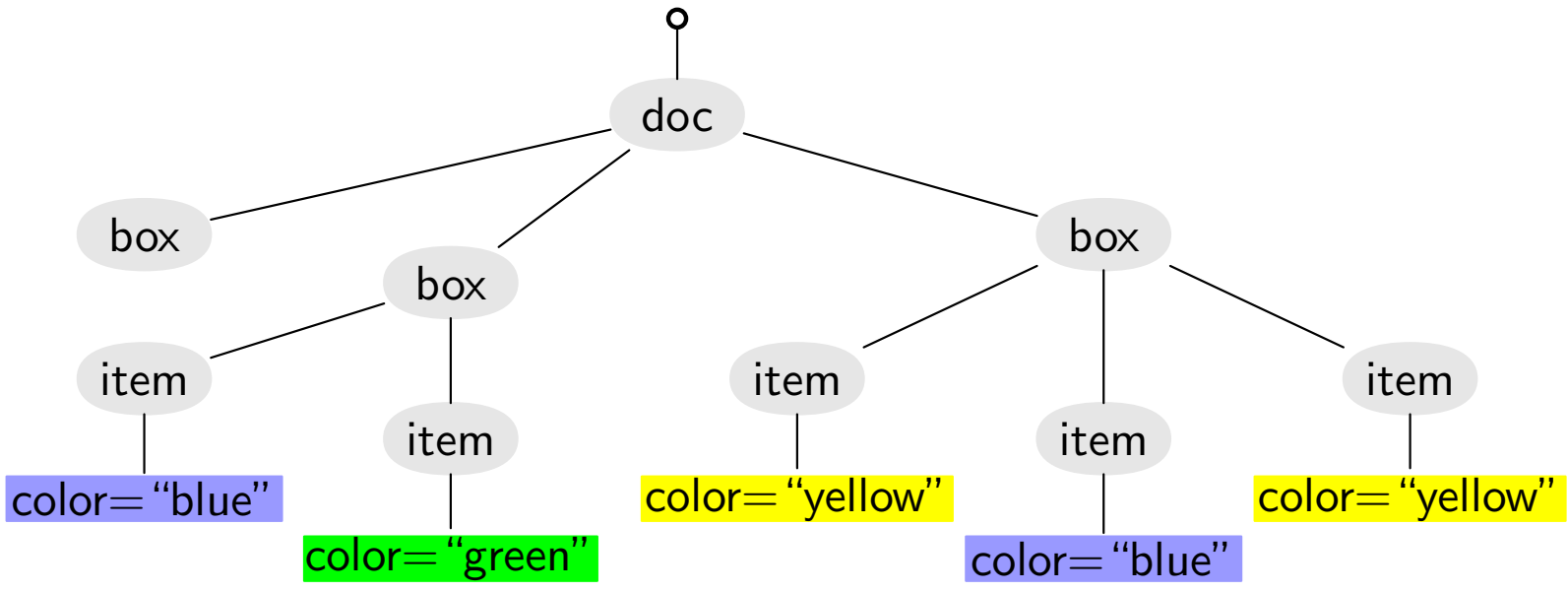
Document:



Query: `//box[item[@color="yellow"]]`

Tricky Abbreviations

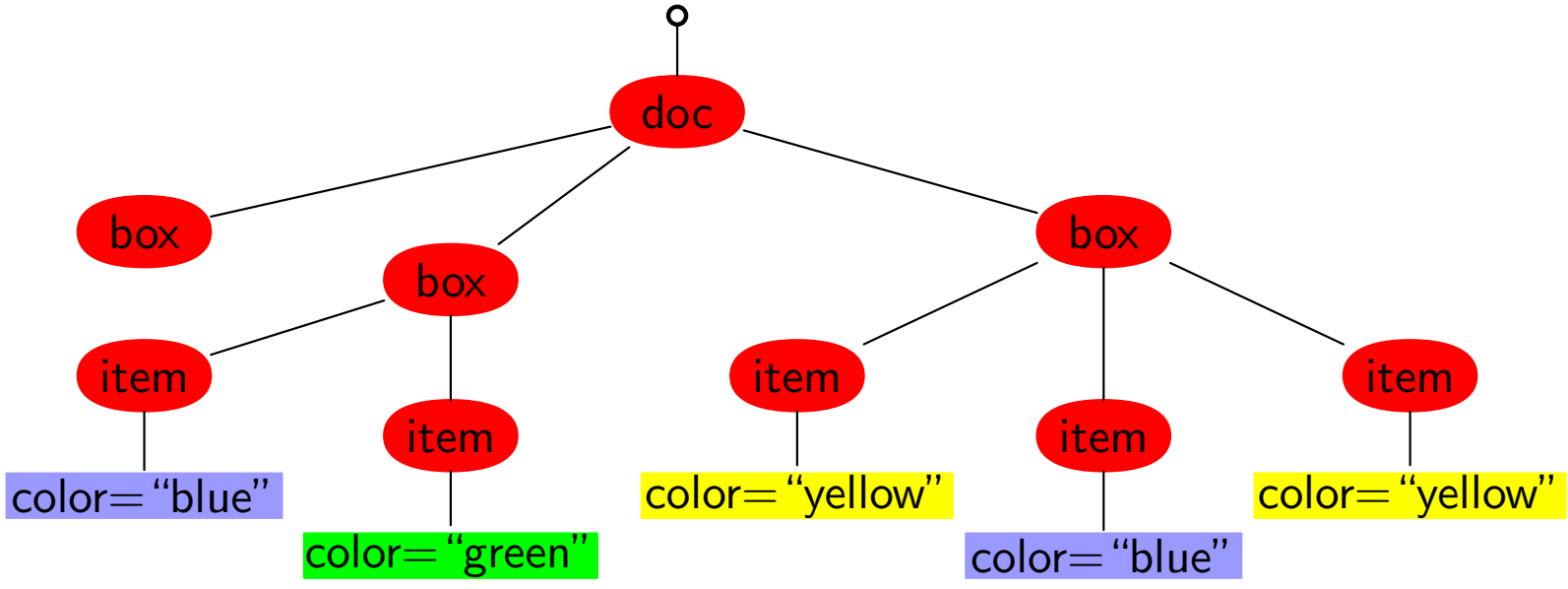
Document:



Query: `/descendant::item[1]`

Tricky Abbreviations

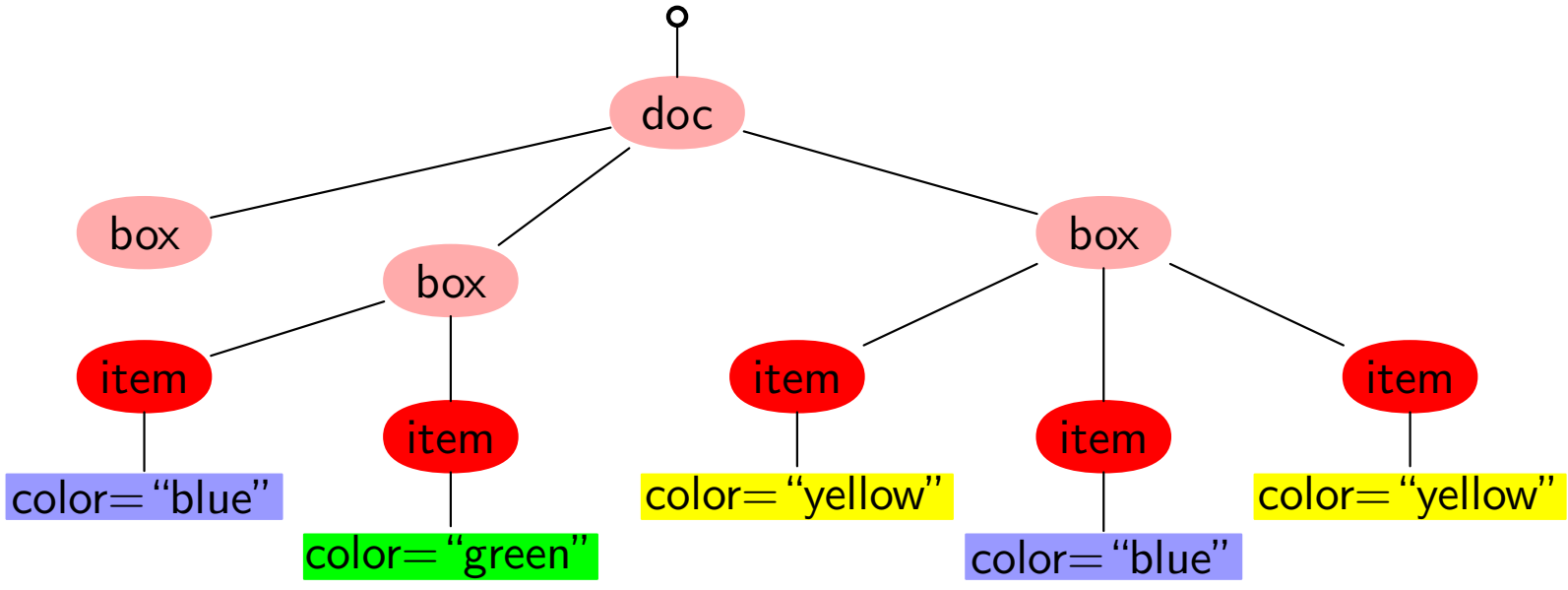
Document:



Query: `/descendant::item[1]`

Tricky Abbreviations

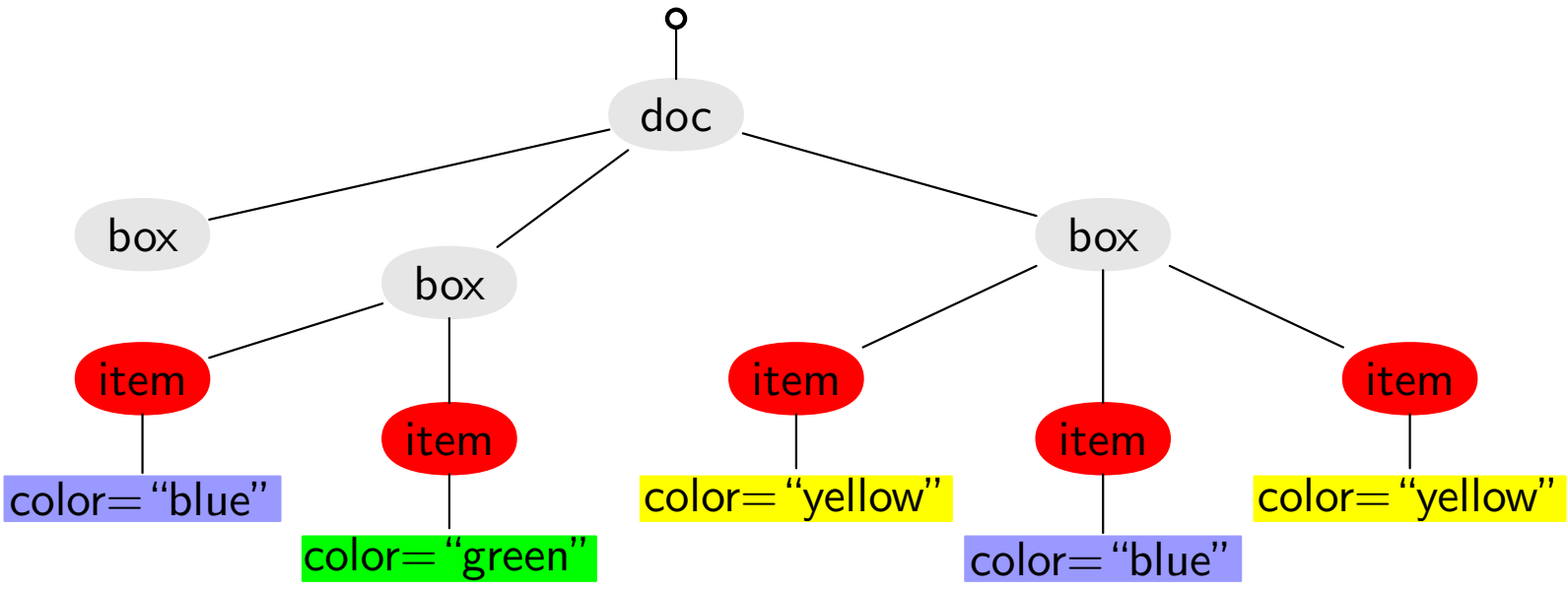
Document:



Query: `/descendant::item[1]`

Tricky Abbreviations

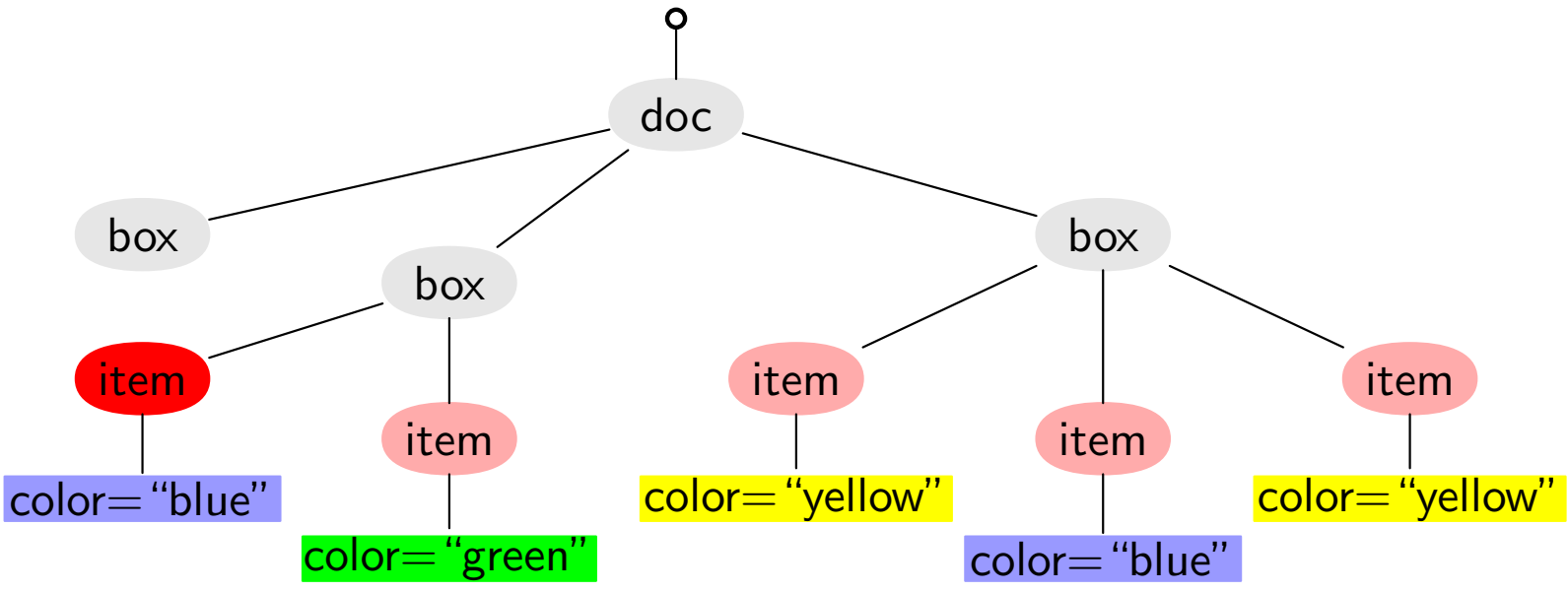
Document:



Query: `/descendant::item[1]`

Tricky Abbreviations

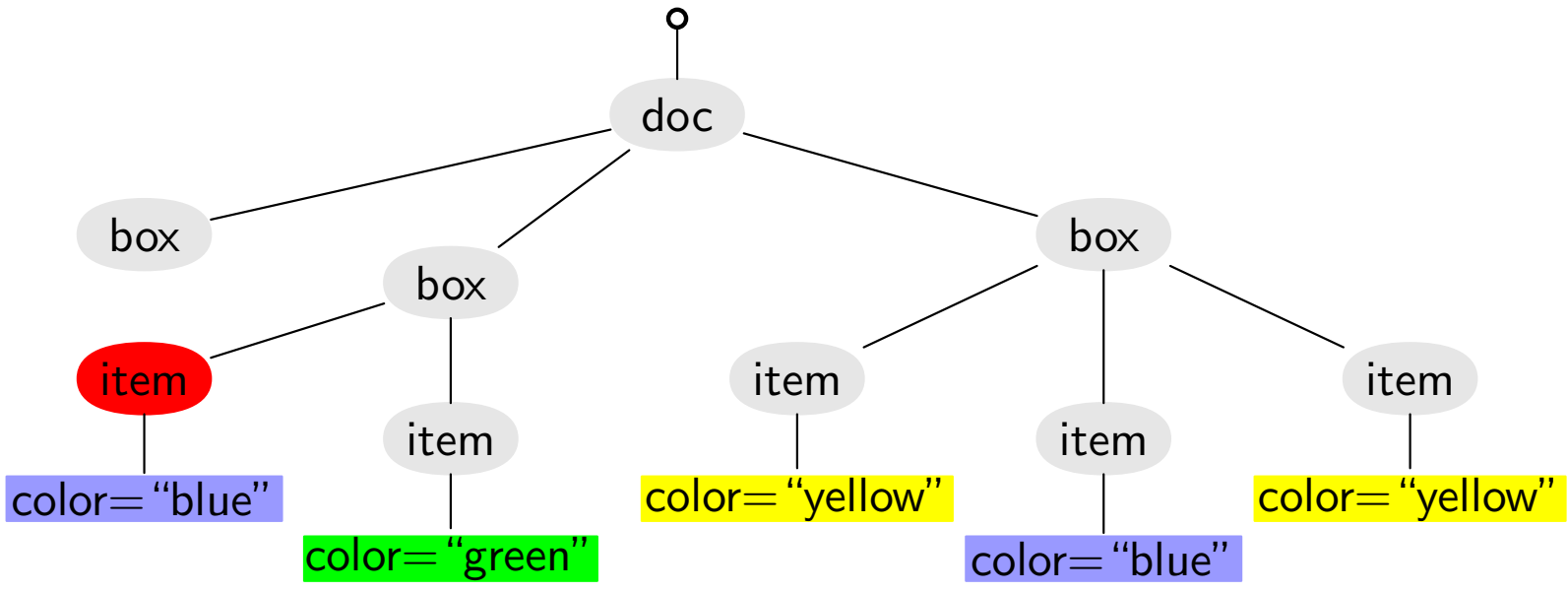
Document:



Query: `/descendant::item[1]`

Tricky Abbreviations

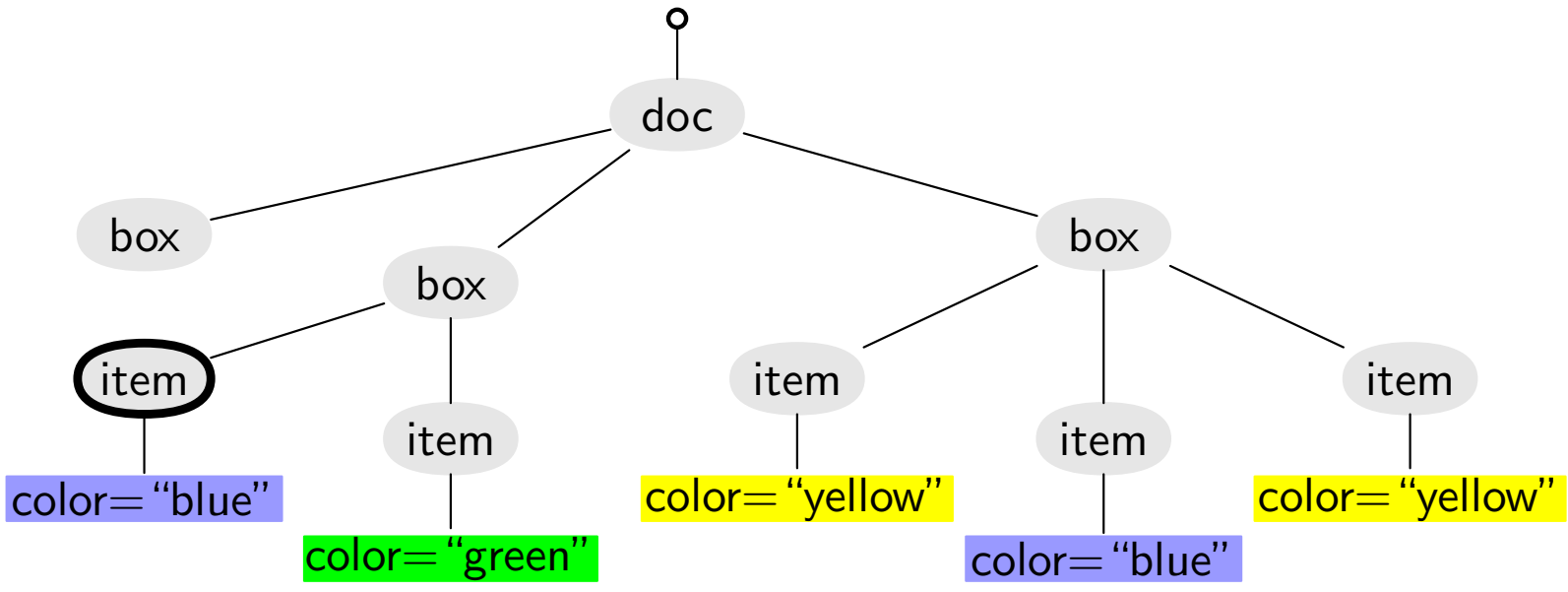
Document:



Query: `/descendant::item[1]`

Tricky Abbreviations

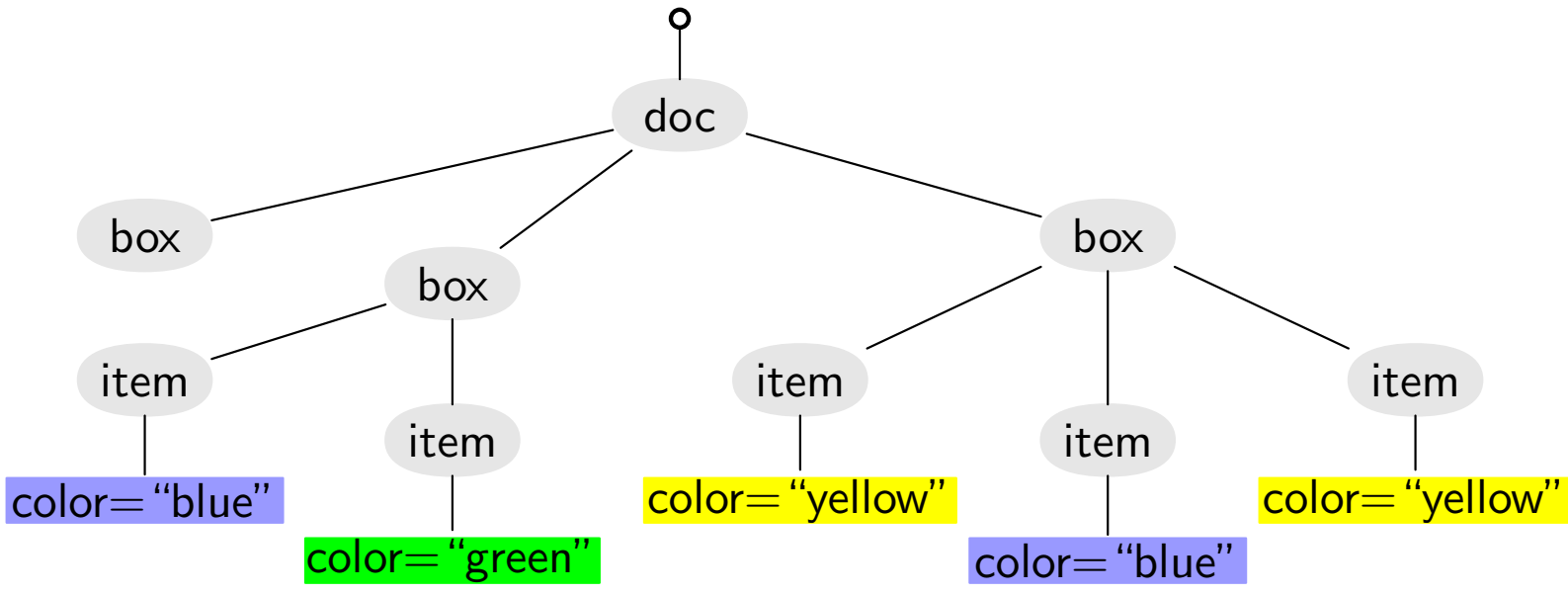
Document:



Query: `/descendant::item[1]`

Tricky Abbreviations

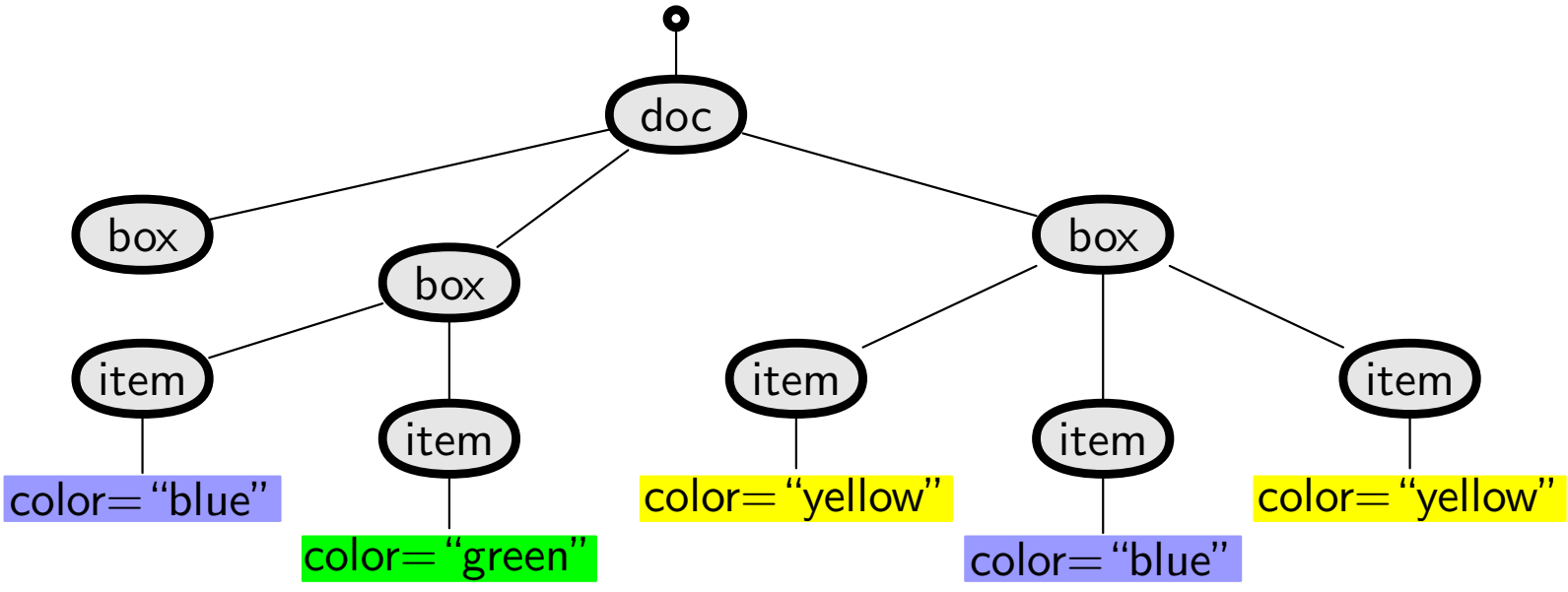
Document:



Query: `//item[1]`

Tricky Abbreviations

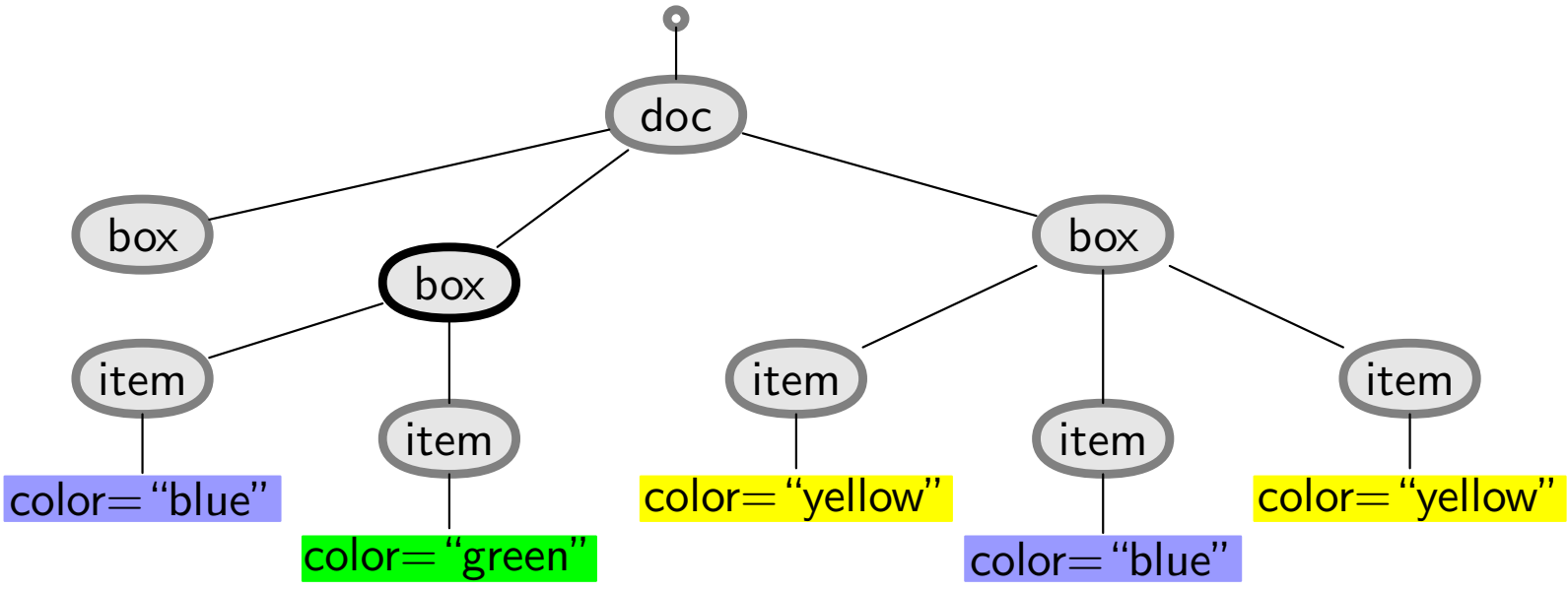
Document:



Query: `//item[1]`

Tricky Abbreviations

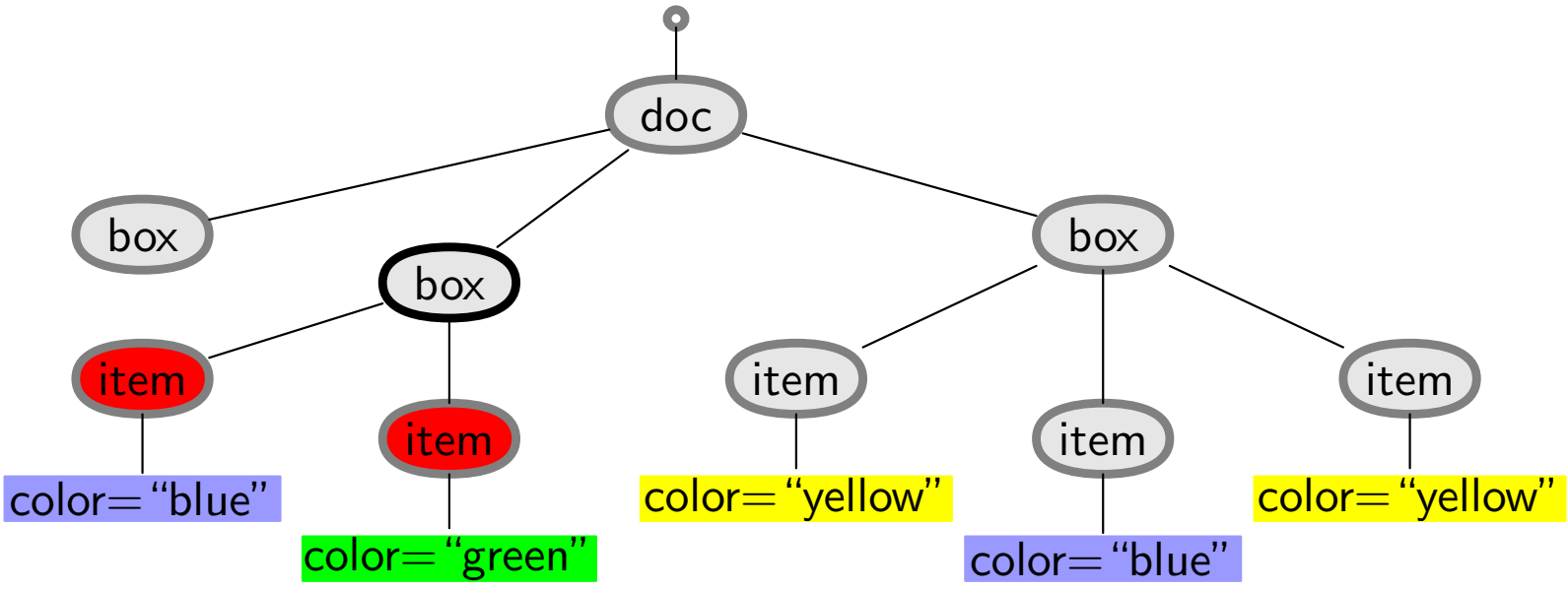
Document:



Query: `//item[1]`

Tricky Abbreviations

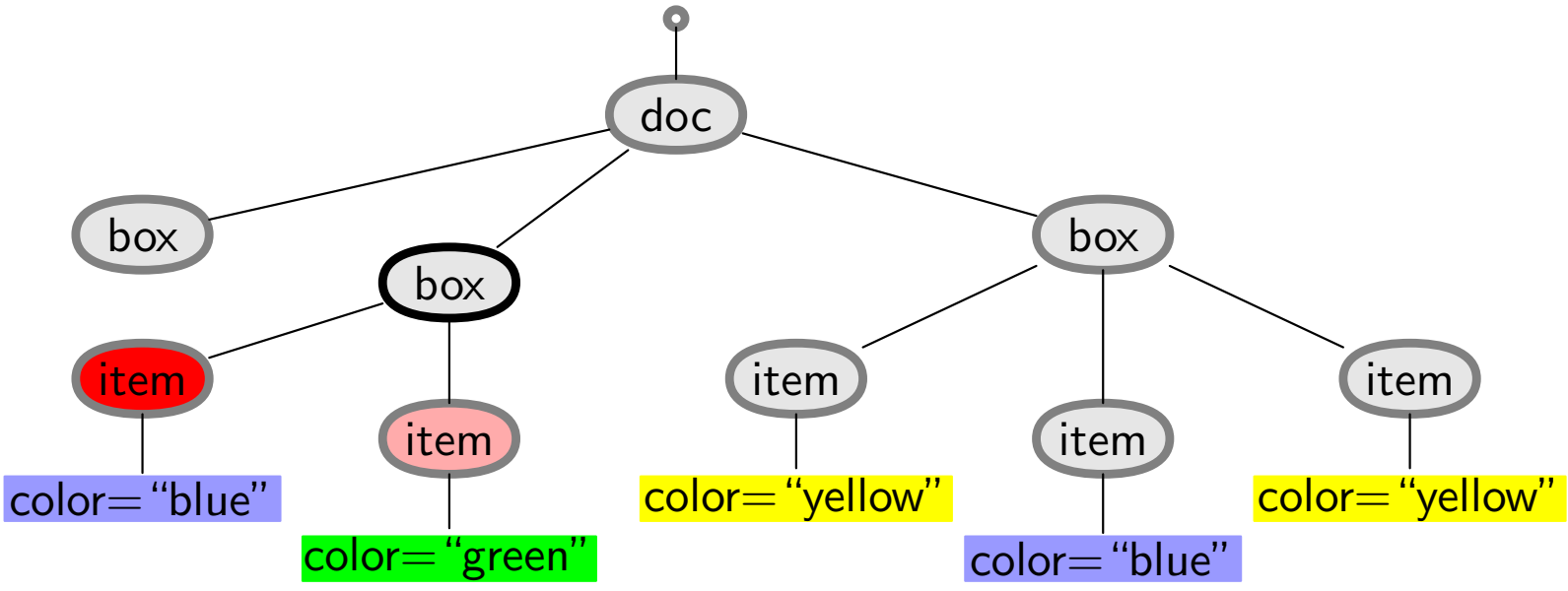
Document:



Query: `//item[1]`

Tricky Abbreviations

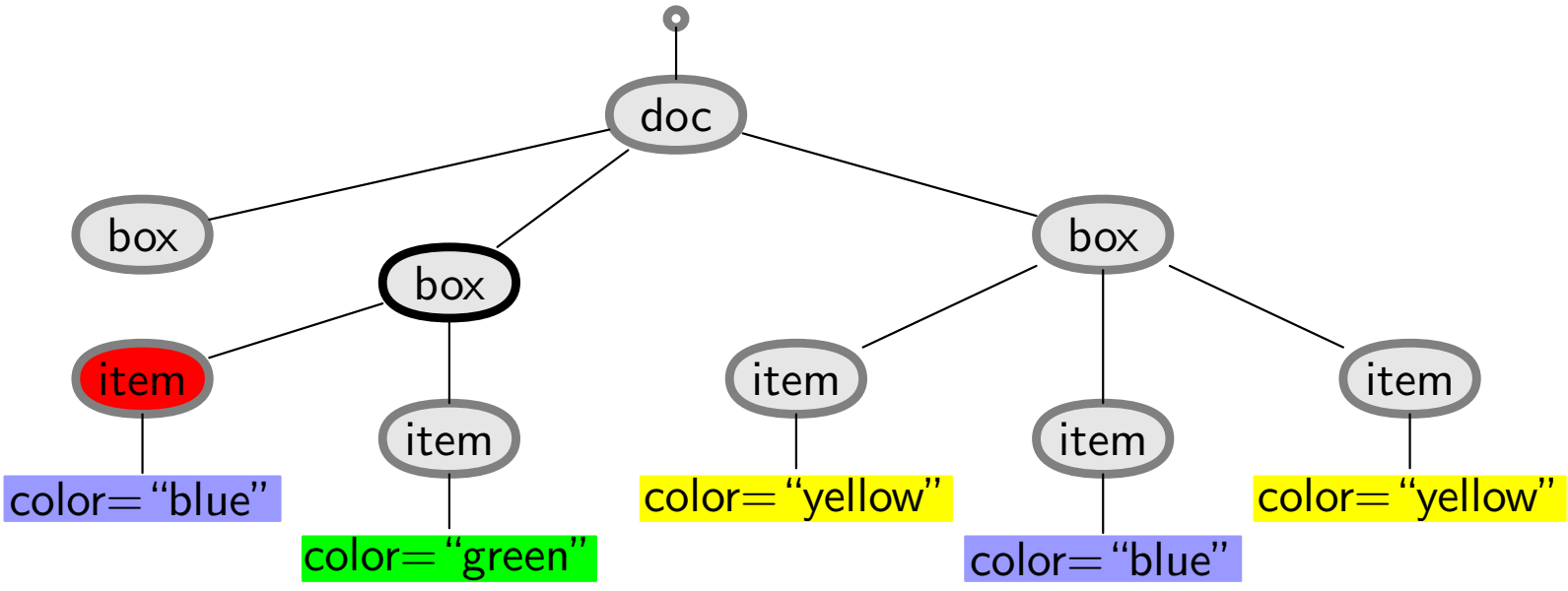
Document:



Query: `//item[1]`

Tricky Abbreviations

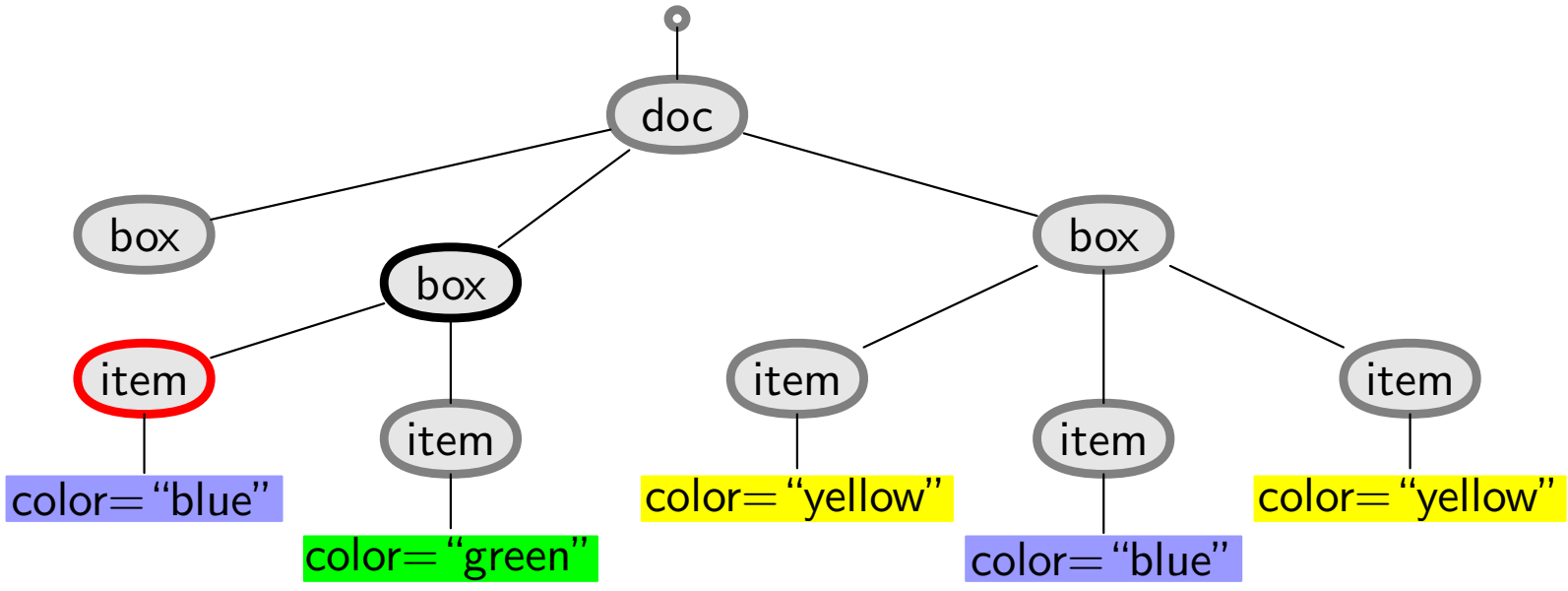
Document:



Query: `//item[1]`

Tricky Abbreviations

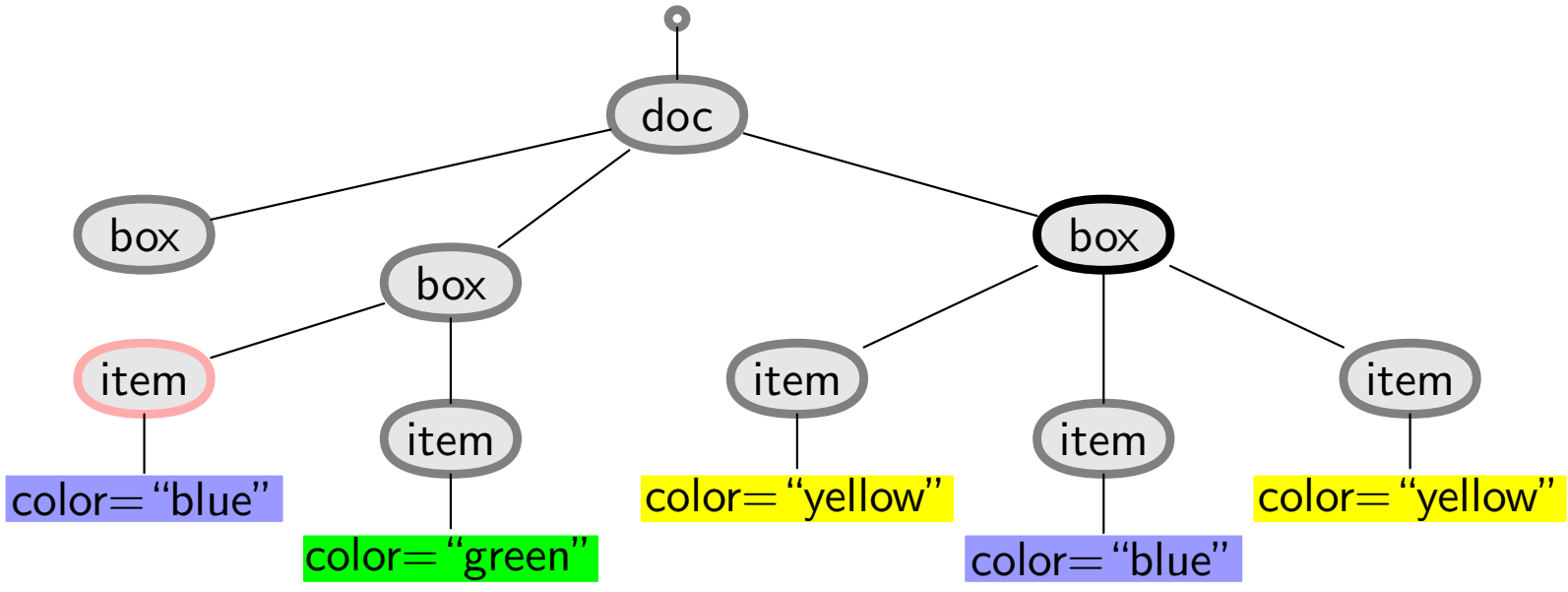
Document:



Query: `//item[1]`

Tricky Abbreviations

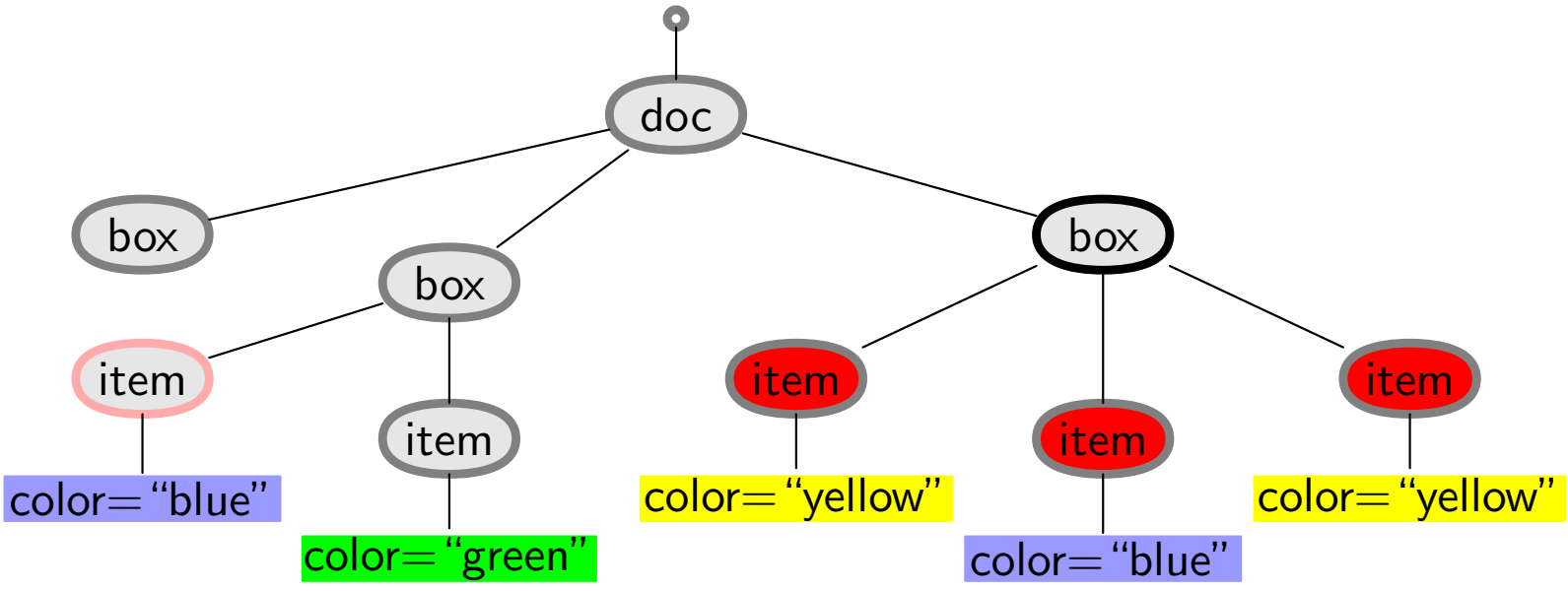
Document:



Query: `//item[1]`

Tricky Abbreviations

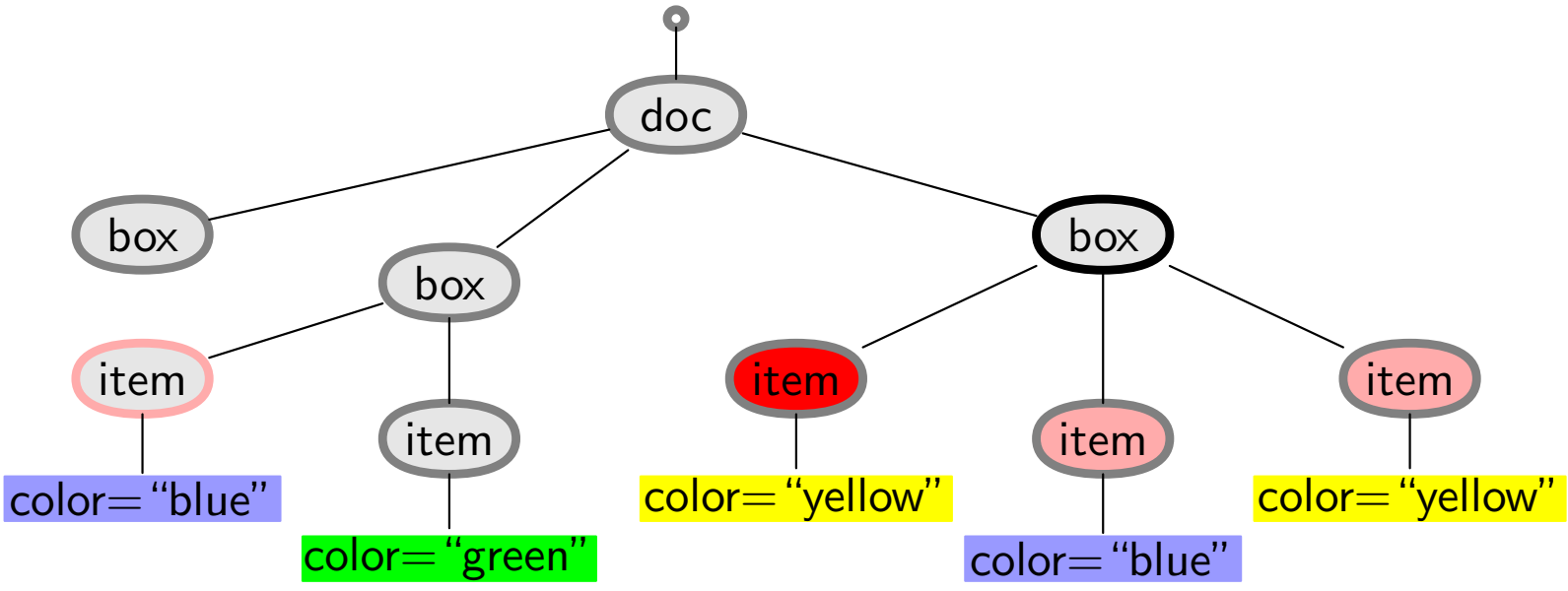
Document:



Query: `//item[1]`

Tricky Abbreviations

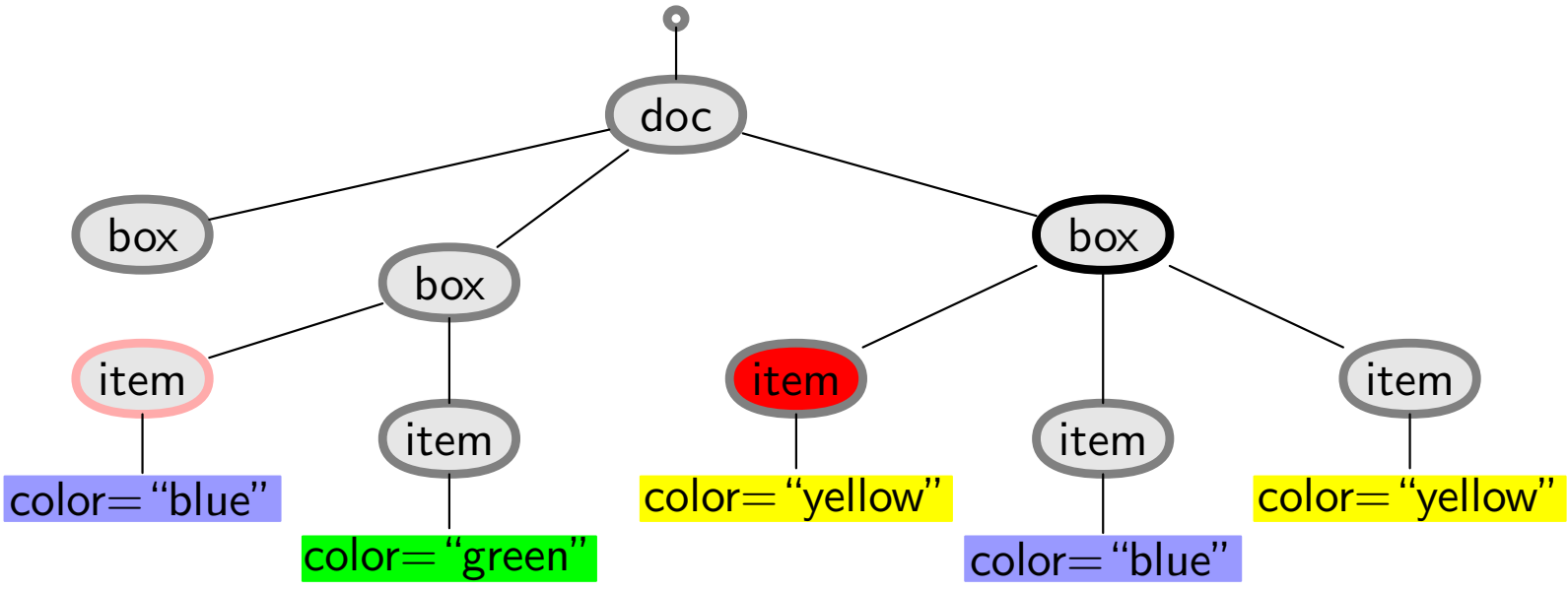
Document:



Query: `//item[1]`

Tricky Abbreviations

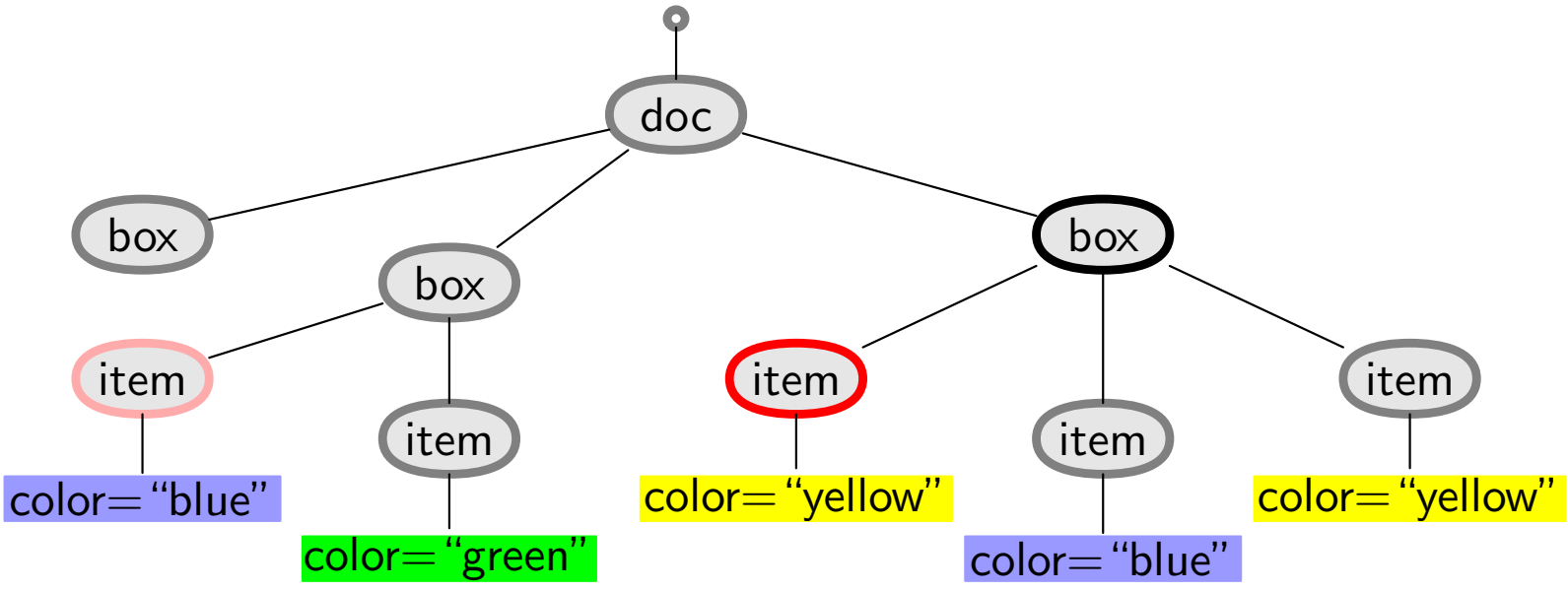
Document:



Query: `//item[1]`

Tricky Abbreviations

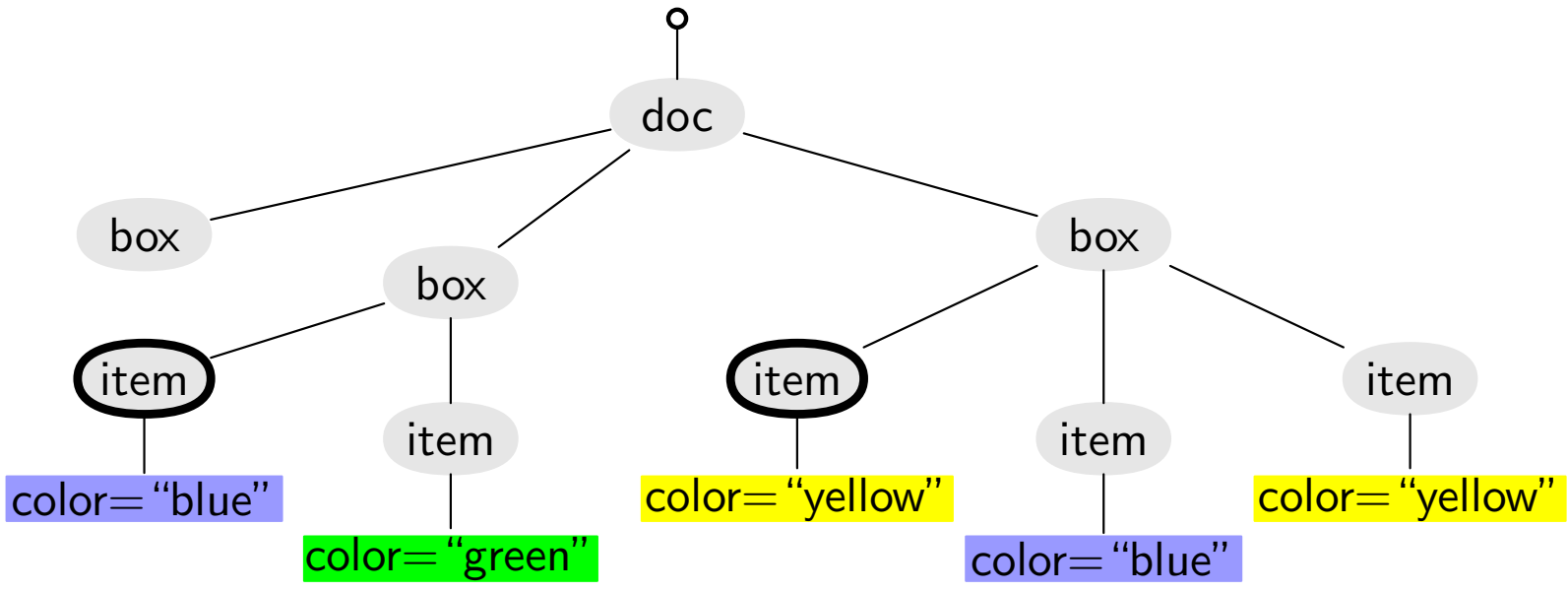
Document:



Query: `//item[1]`

Tricky Abbreviations

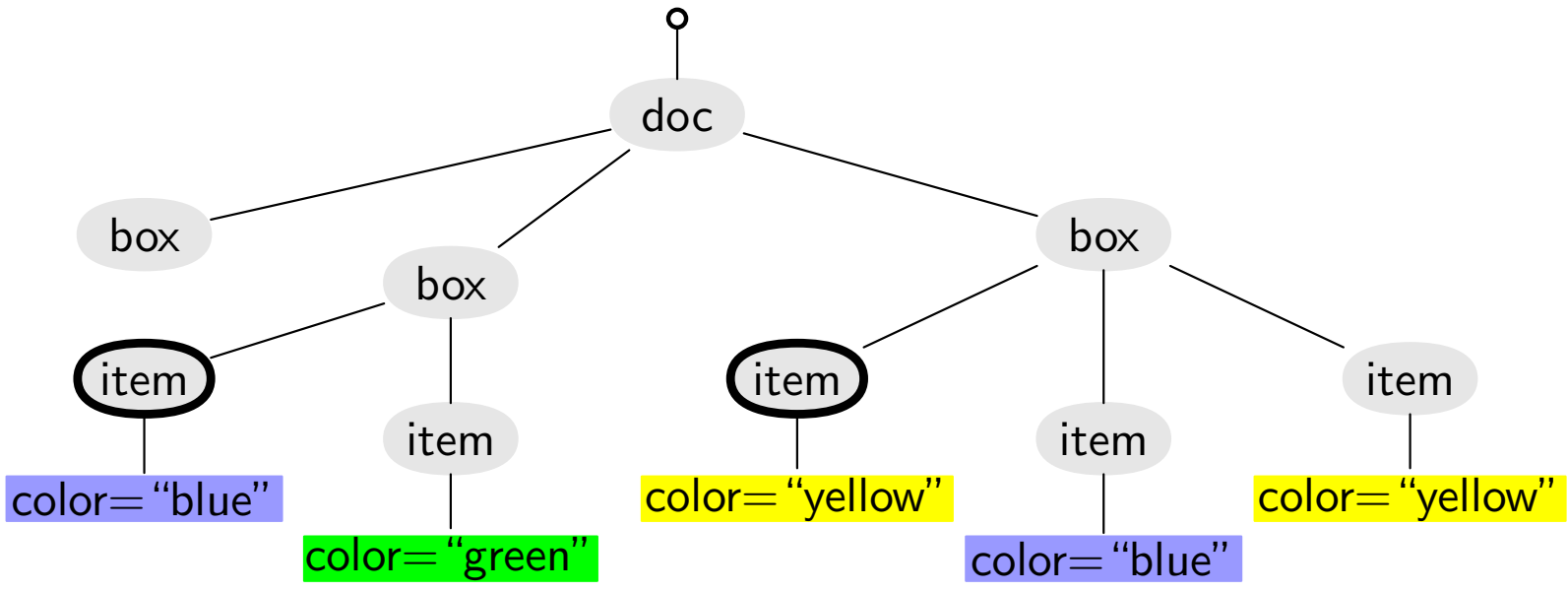
Document:



Query: `//item[1]`

Tricky Abbreviations

Document:

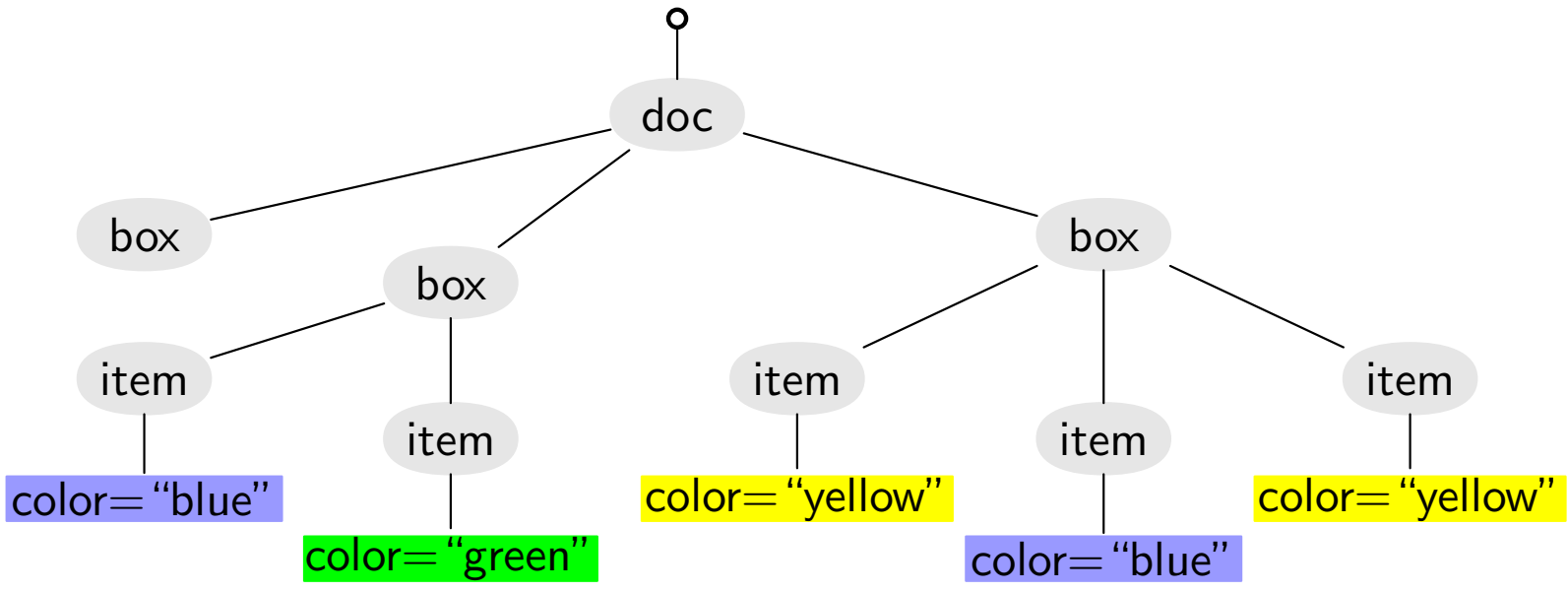


Query: `//item[1]`

Note the difference to `/descendant::item[1]` !

Commutation of Filter Expressions

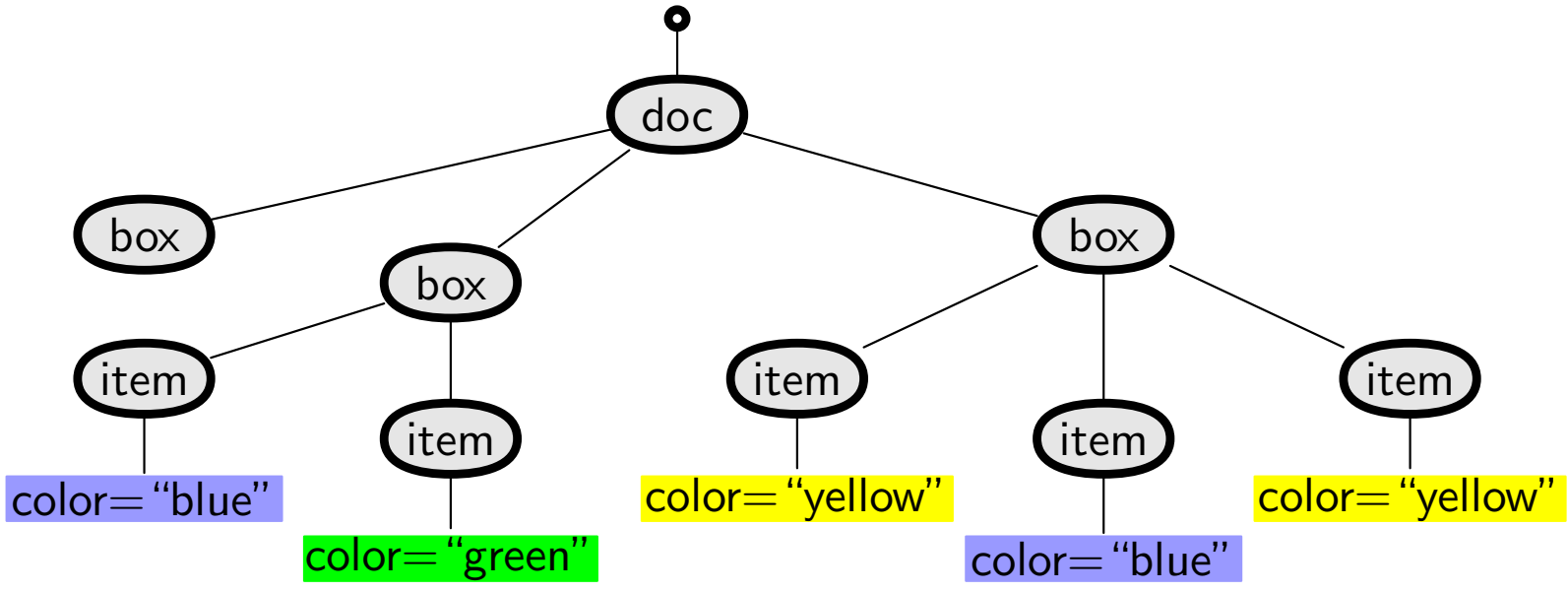
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

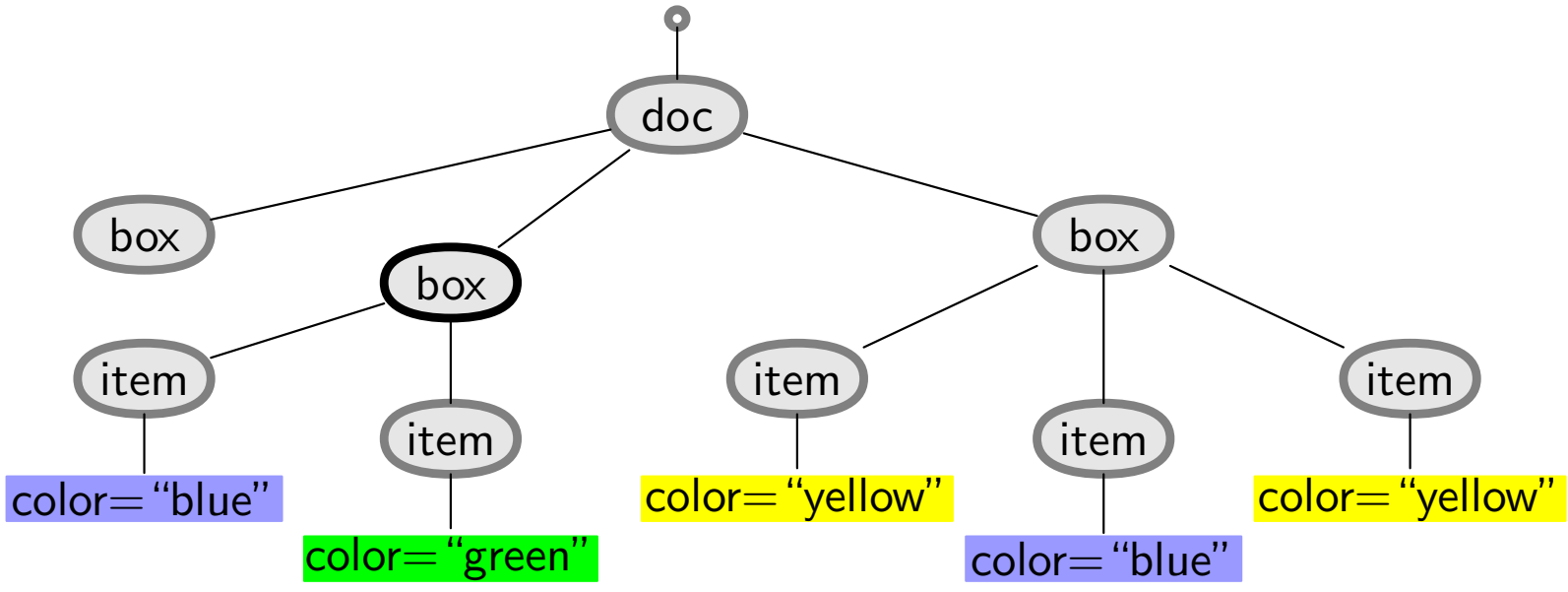
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

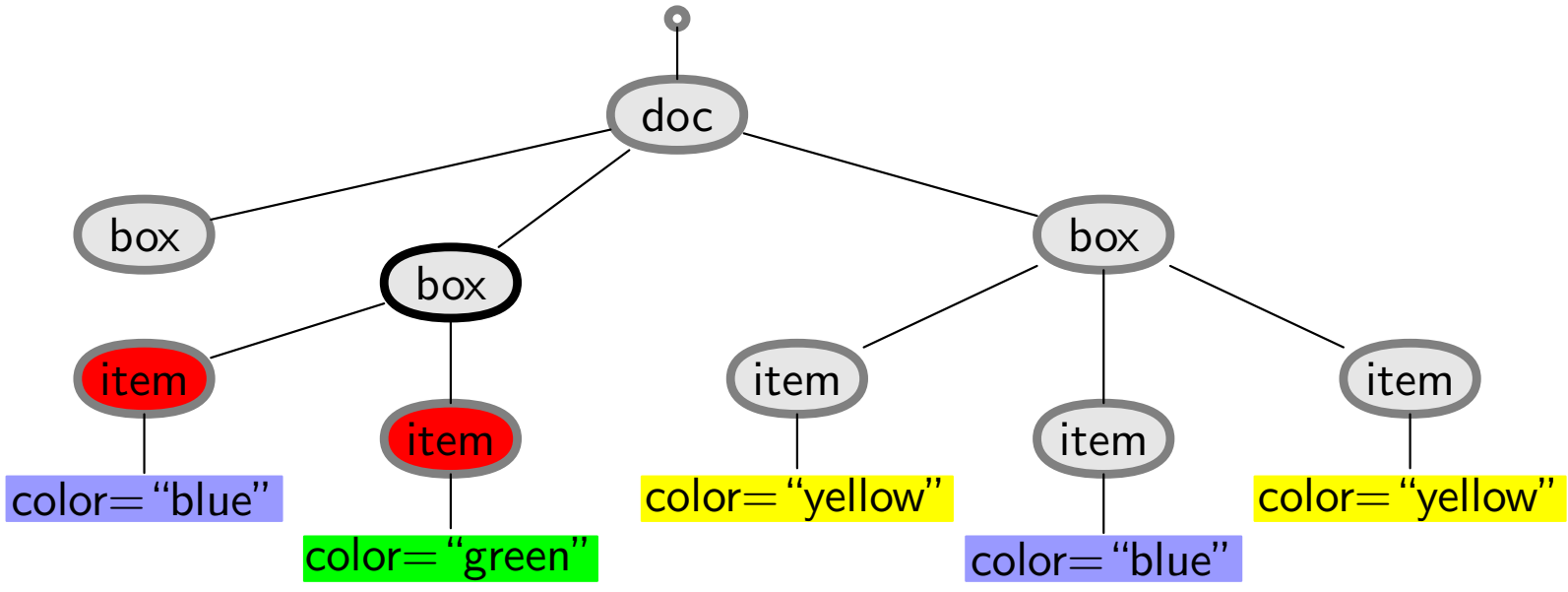
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

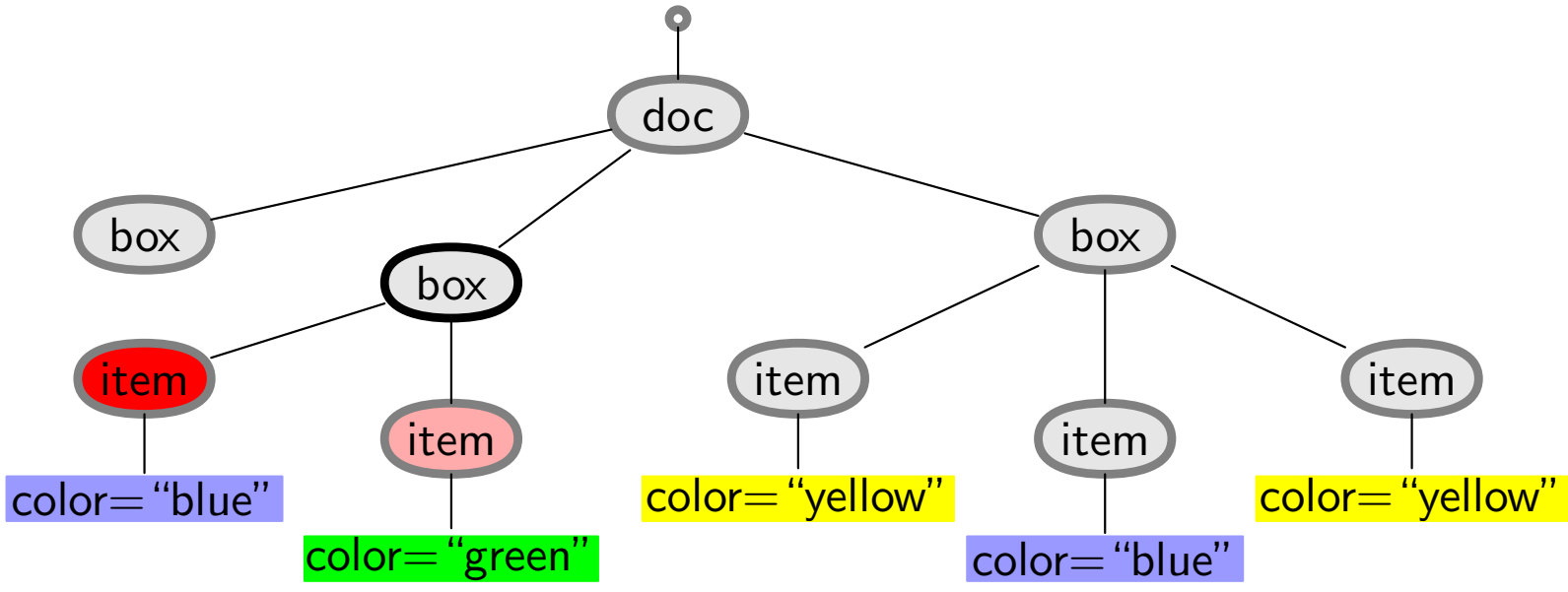
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

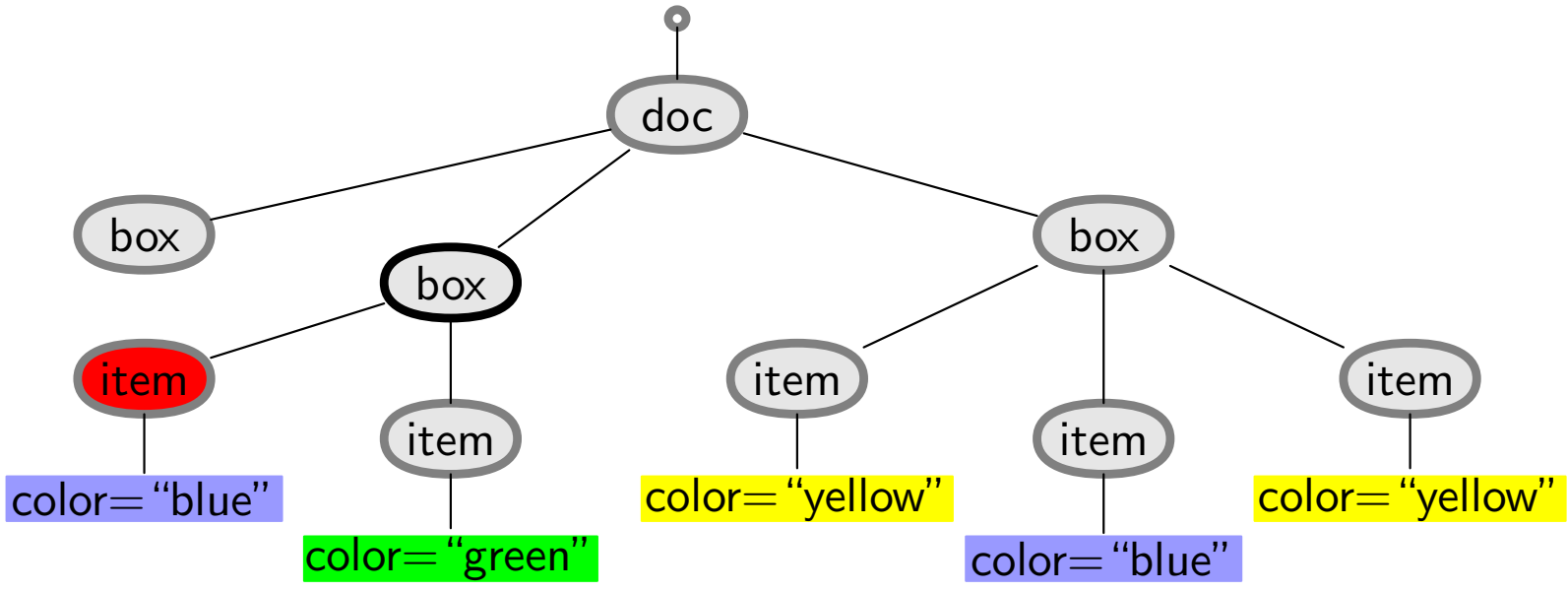
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

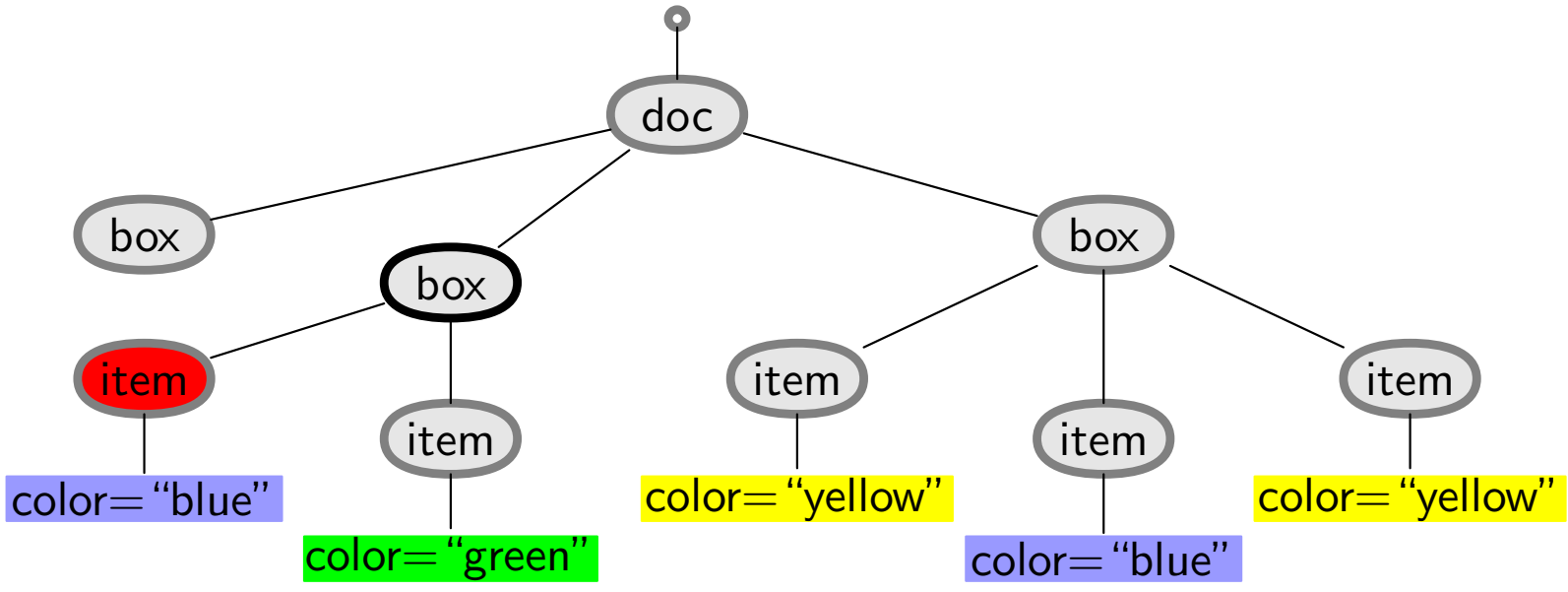
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

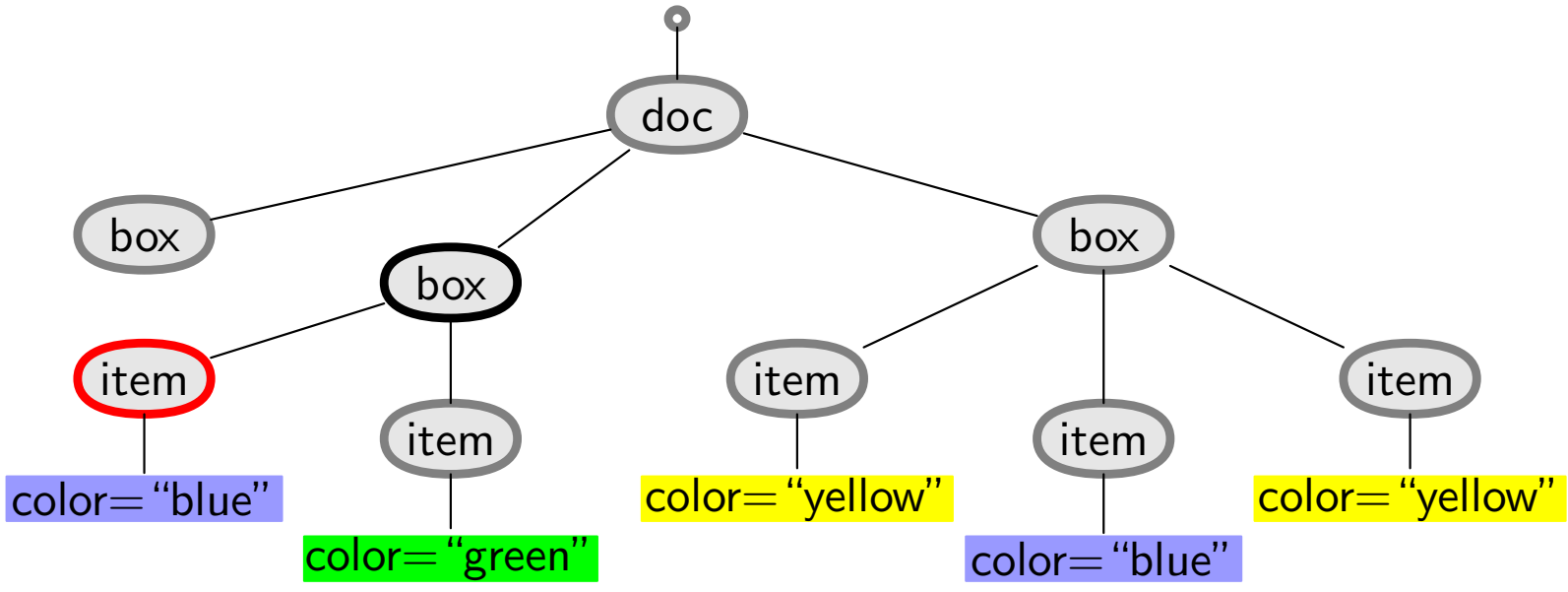
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

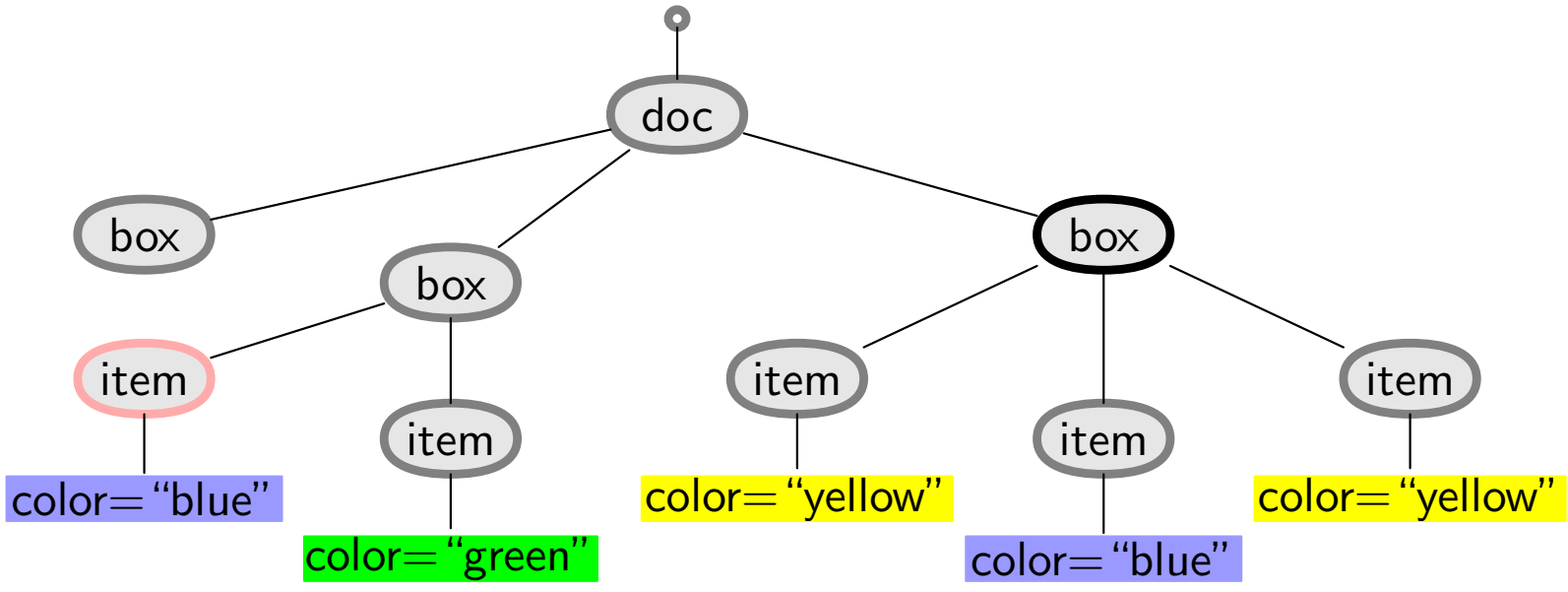
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

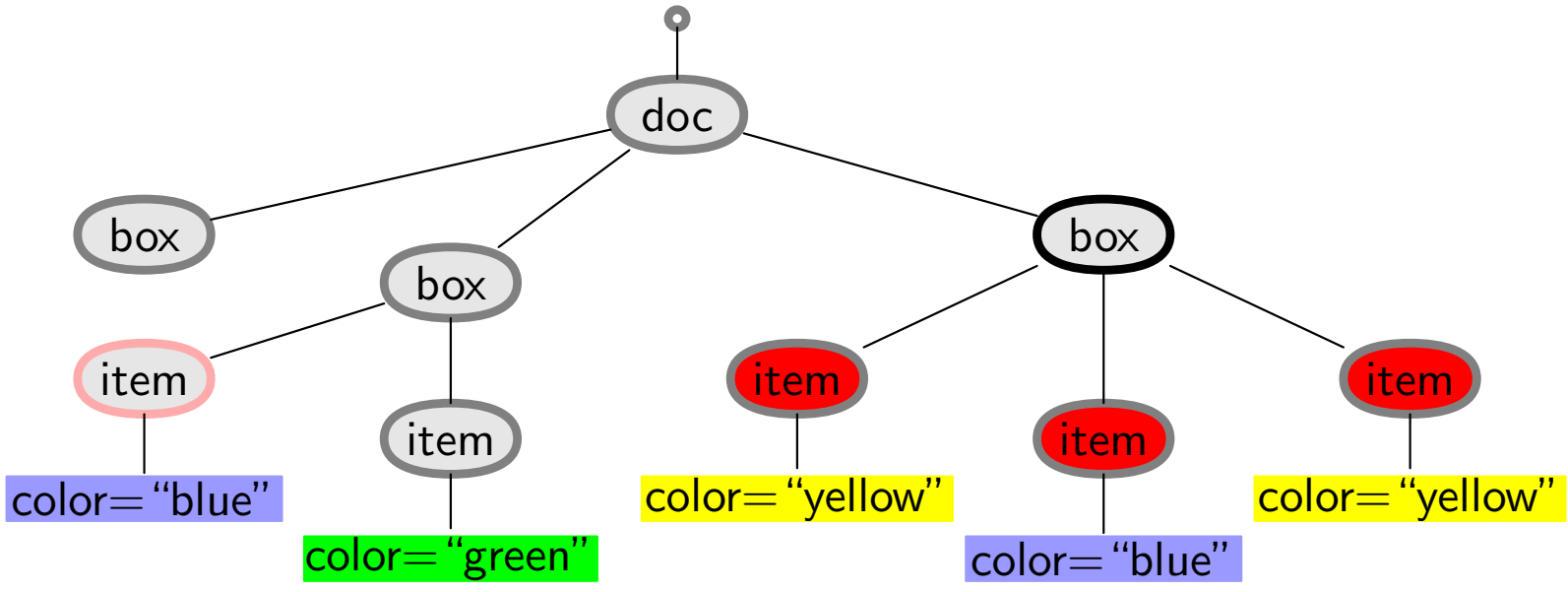
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

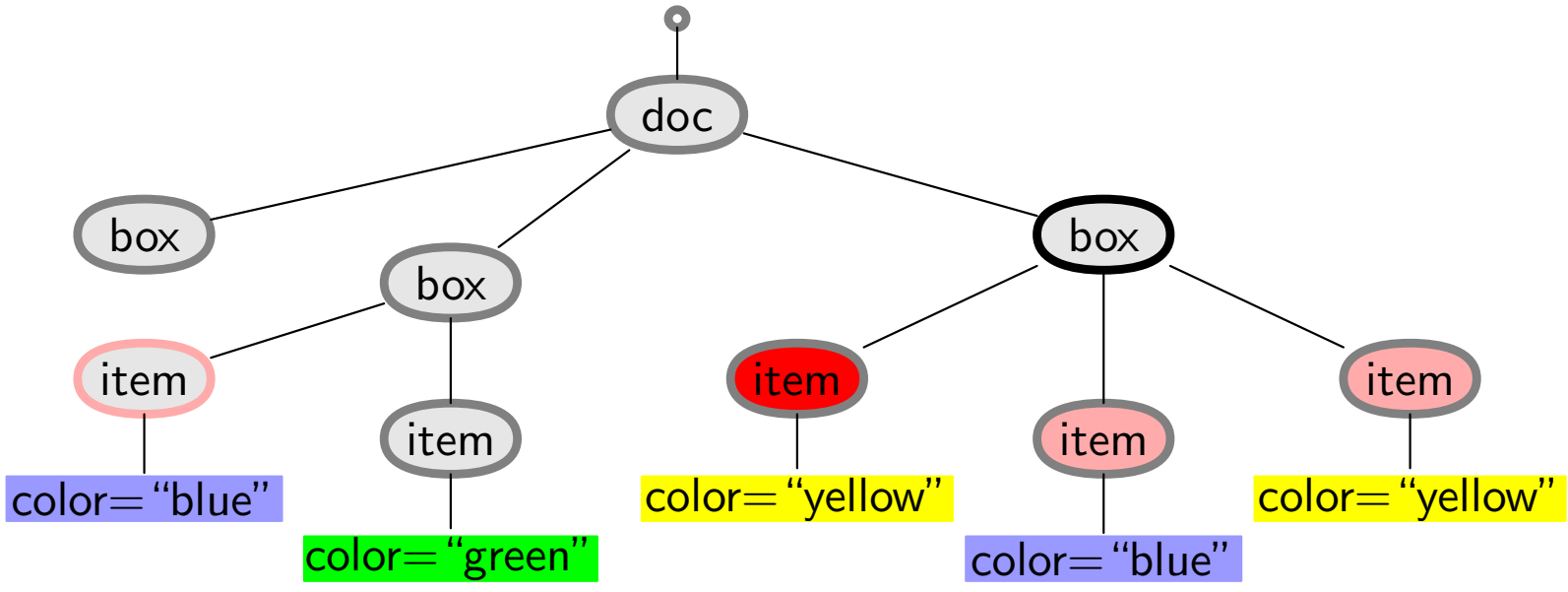
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

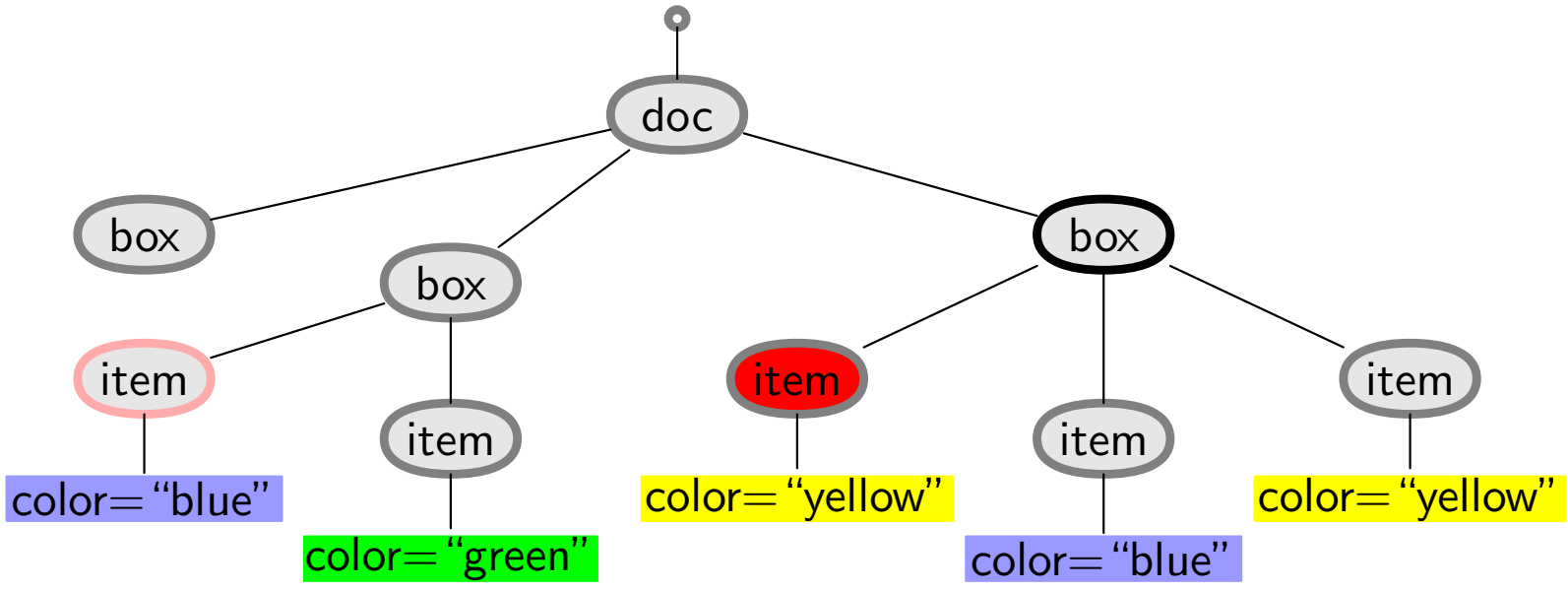
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

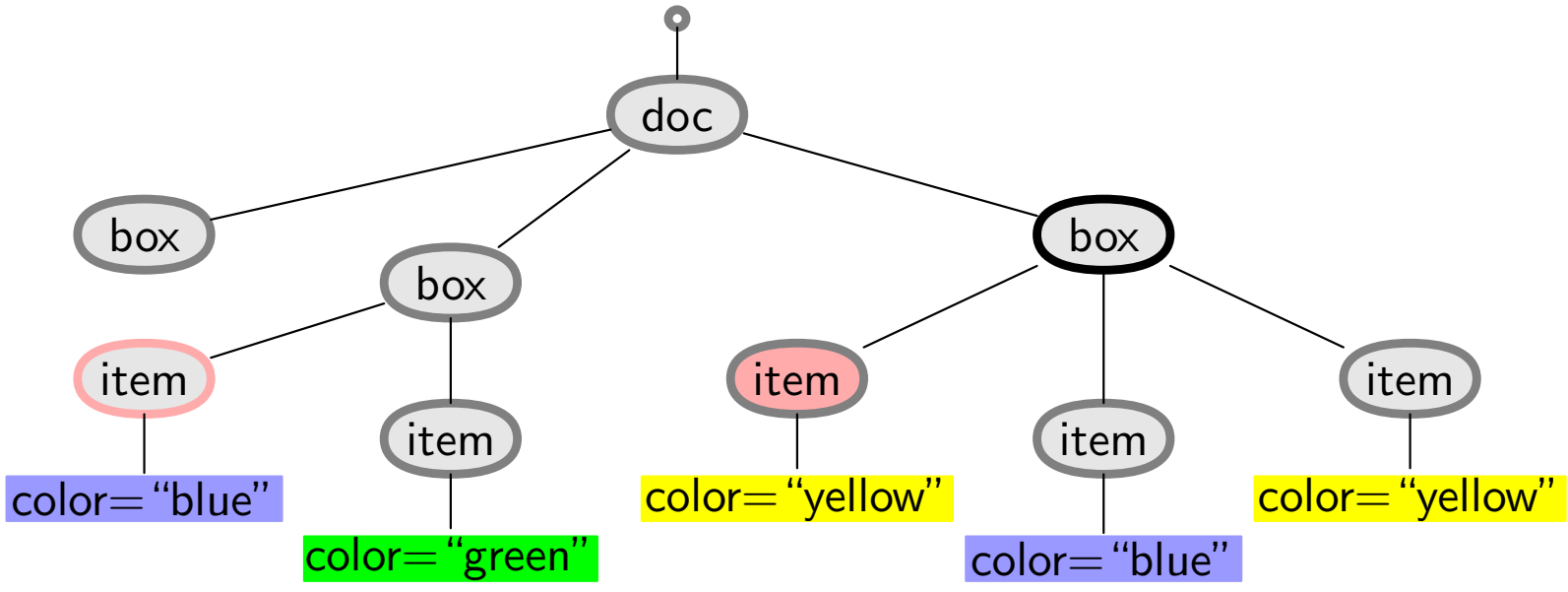
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

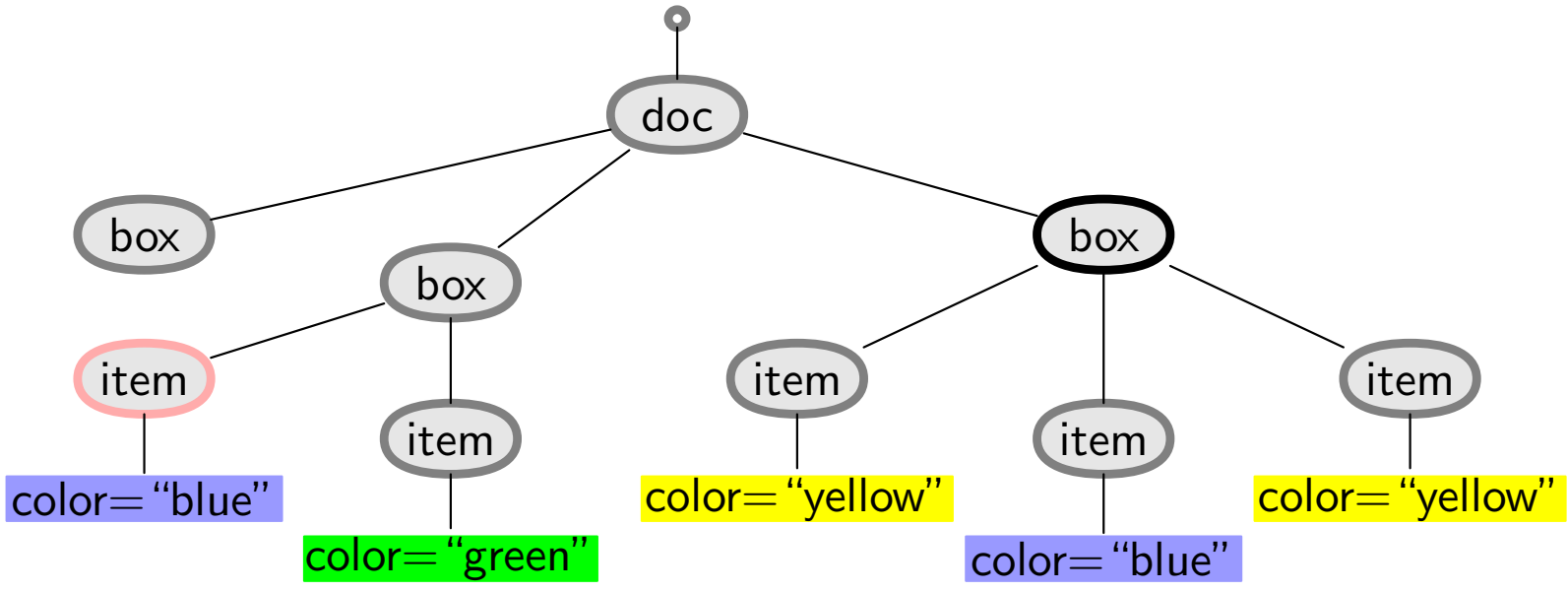
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

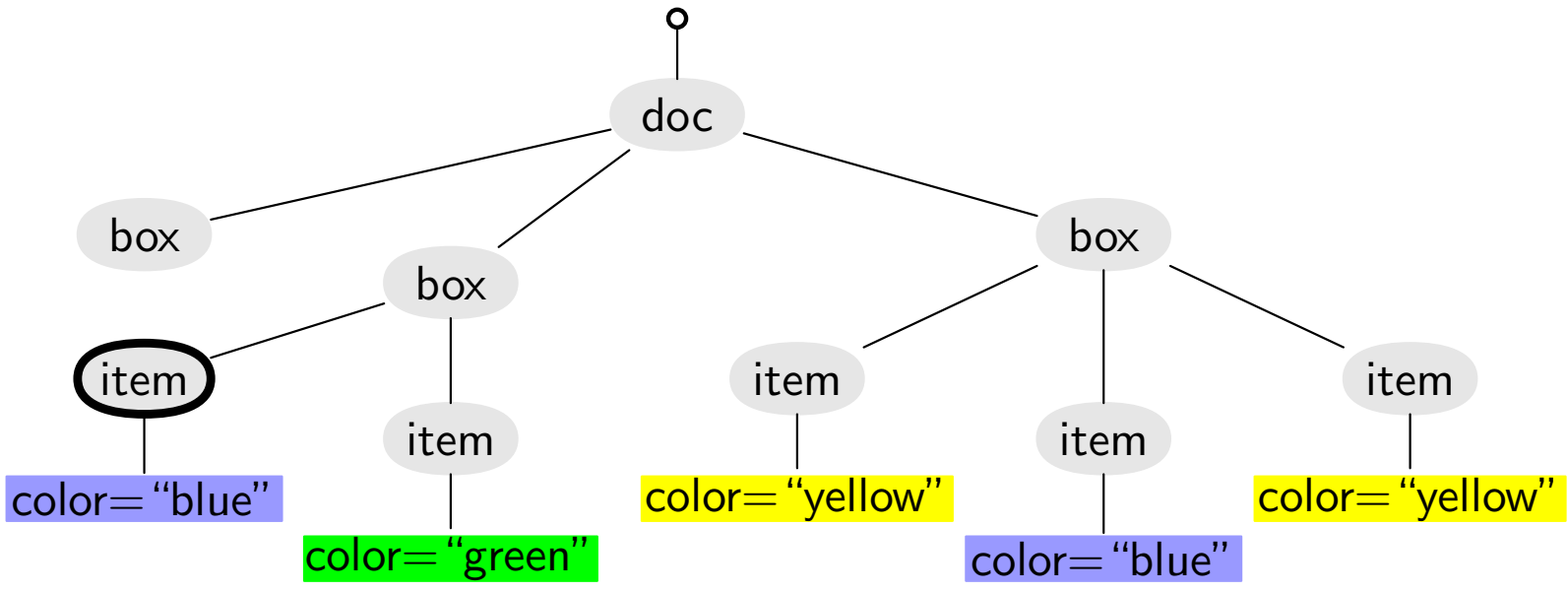
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

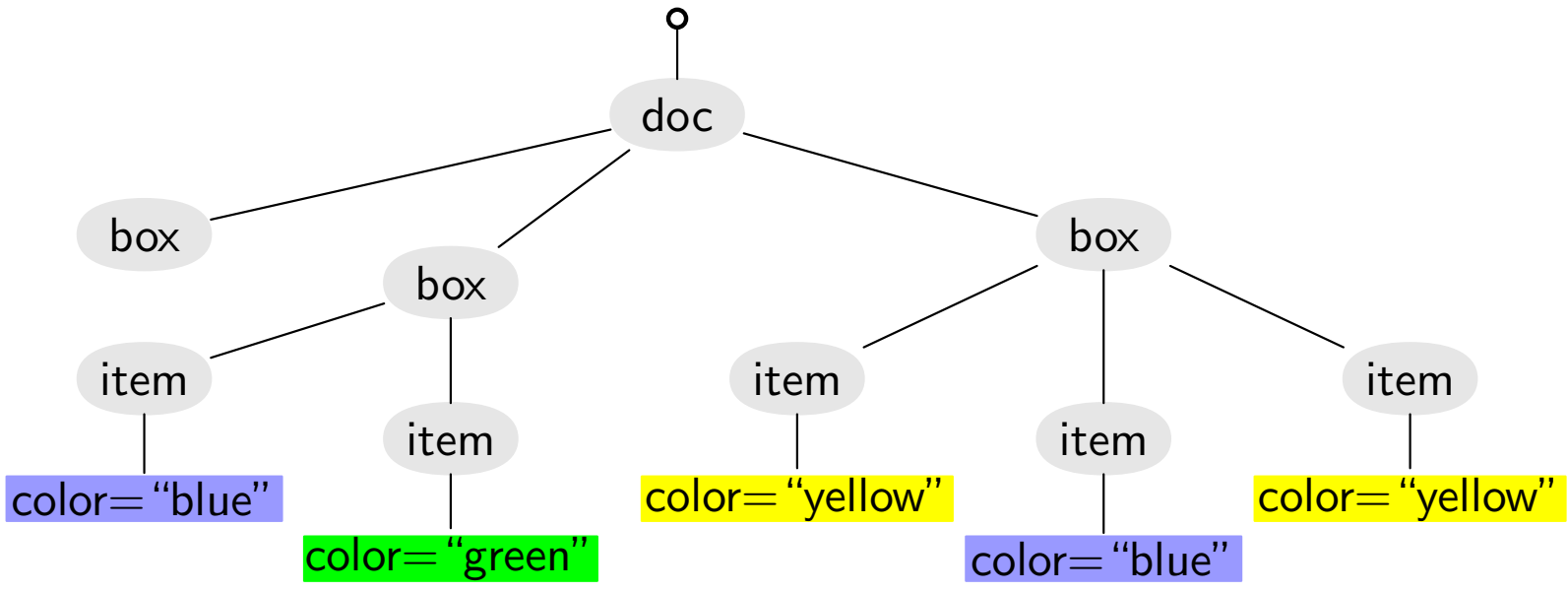
Document:



Query: `//item[1][@color="blue"]`

Commutation of Filter Expressions

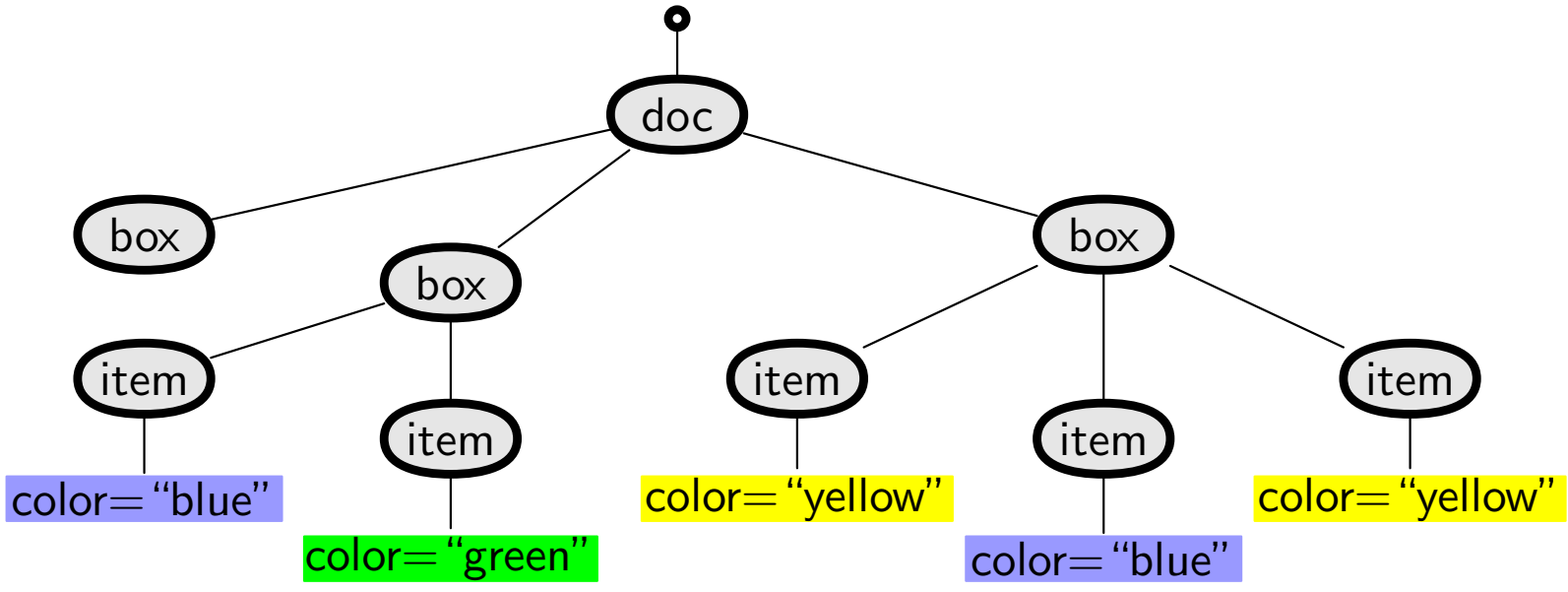
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

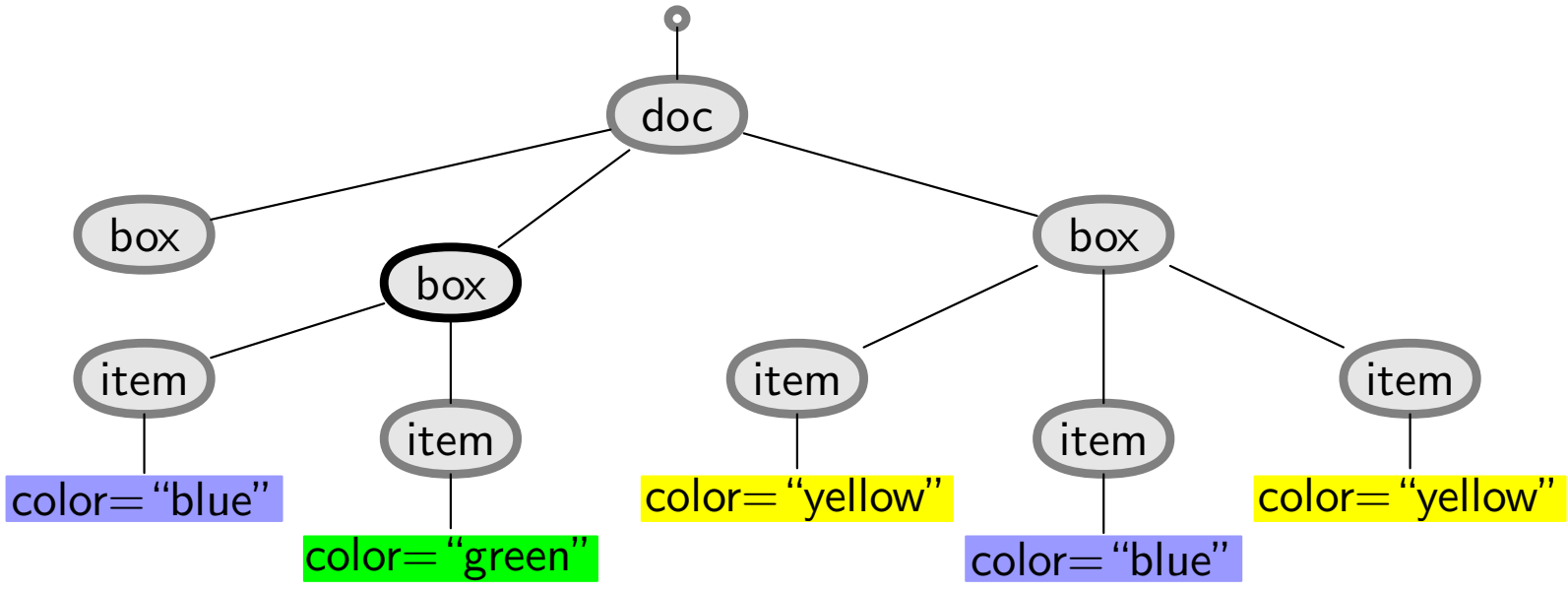
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

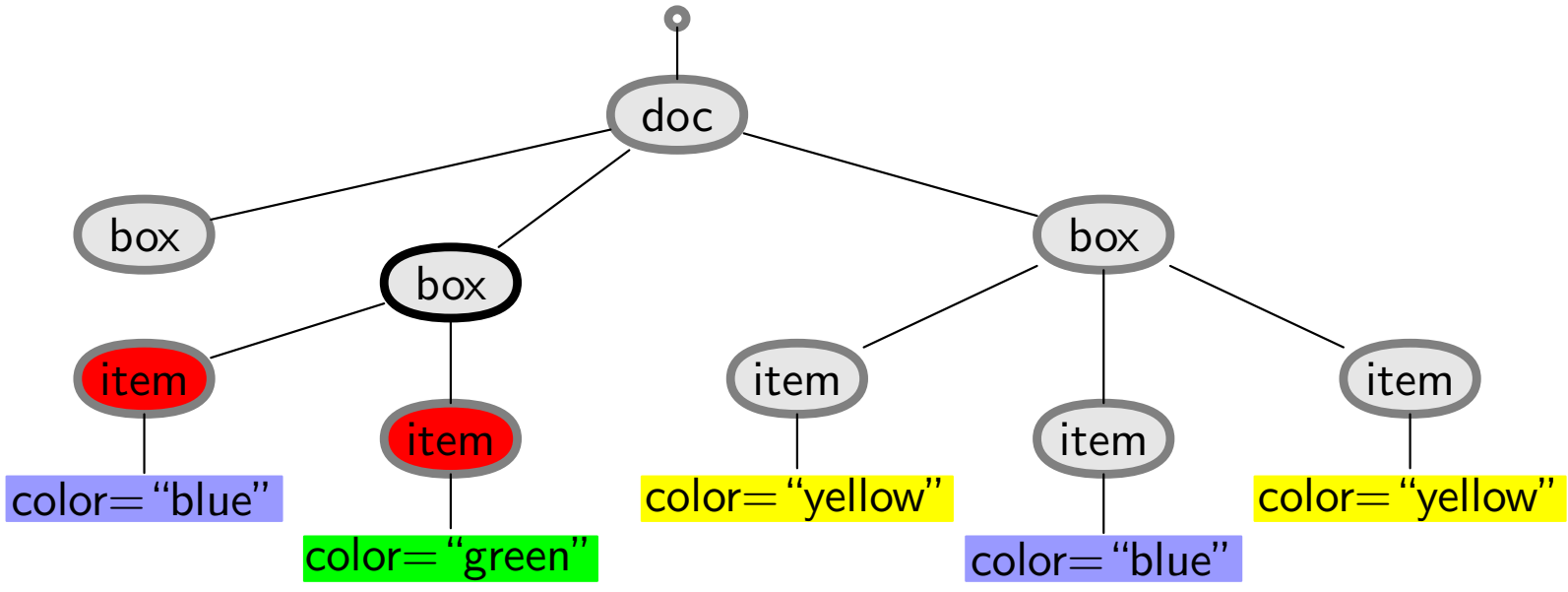
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

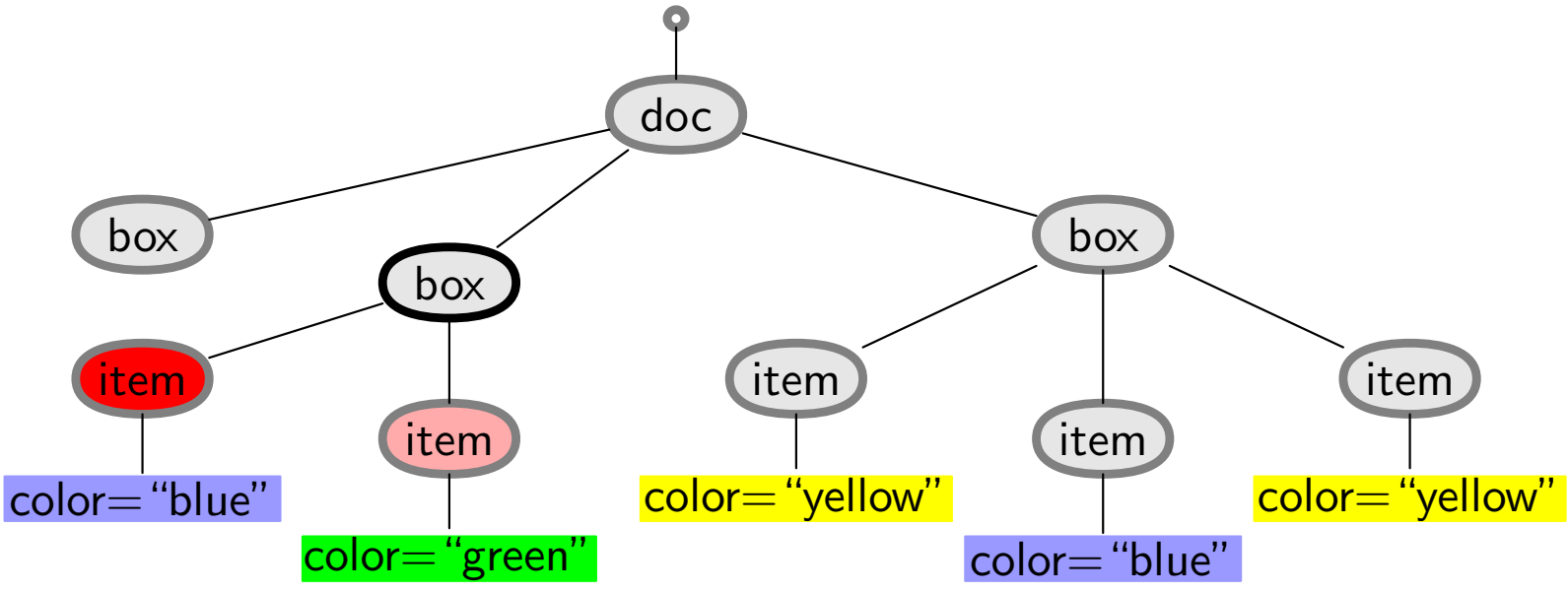
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

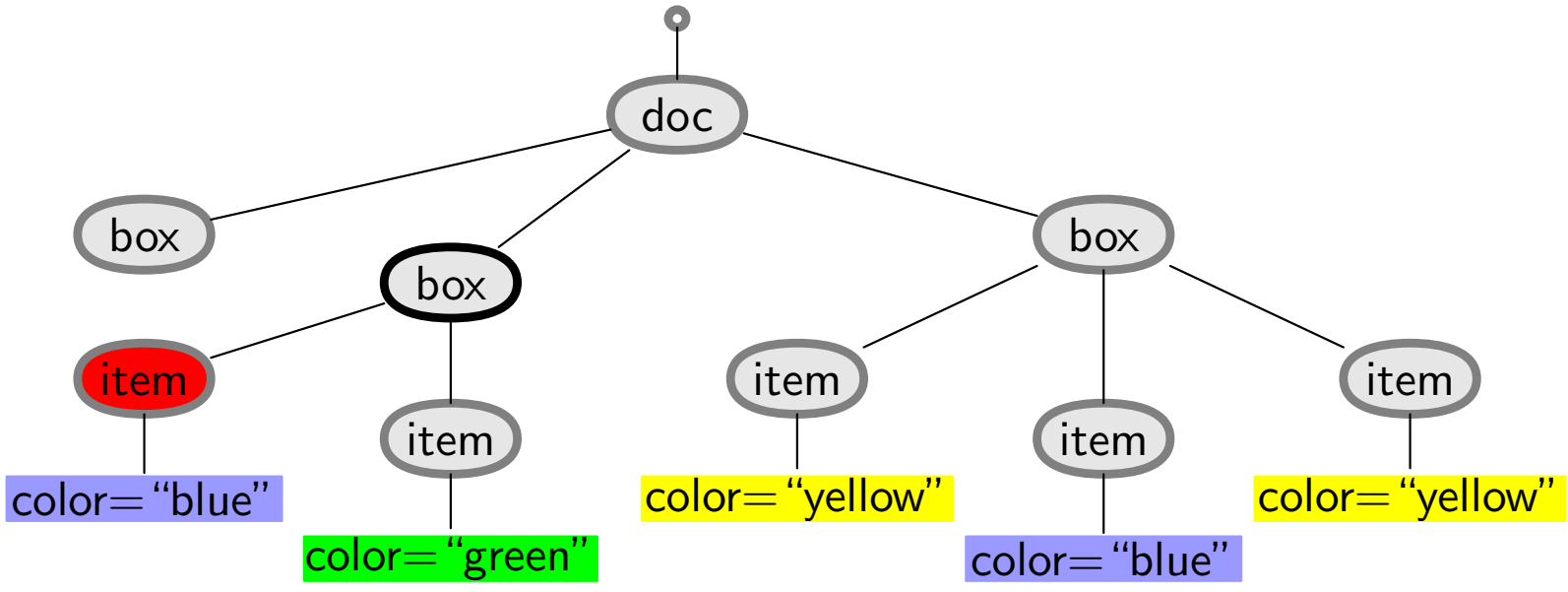
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

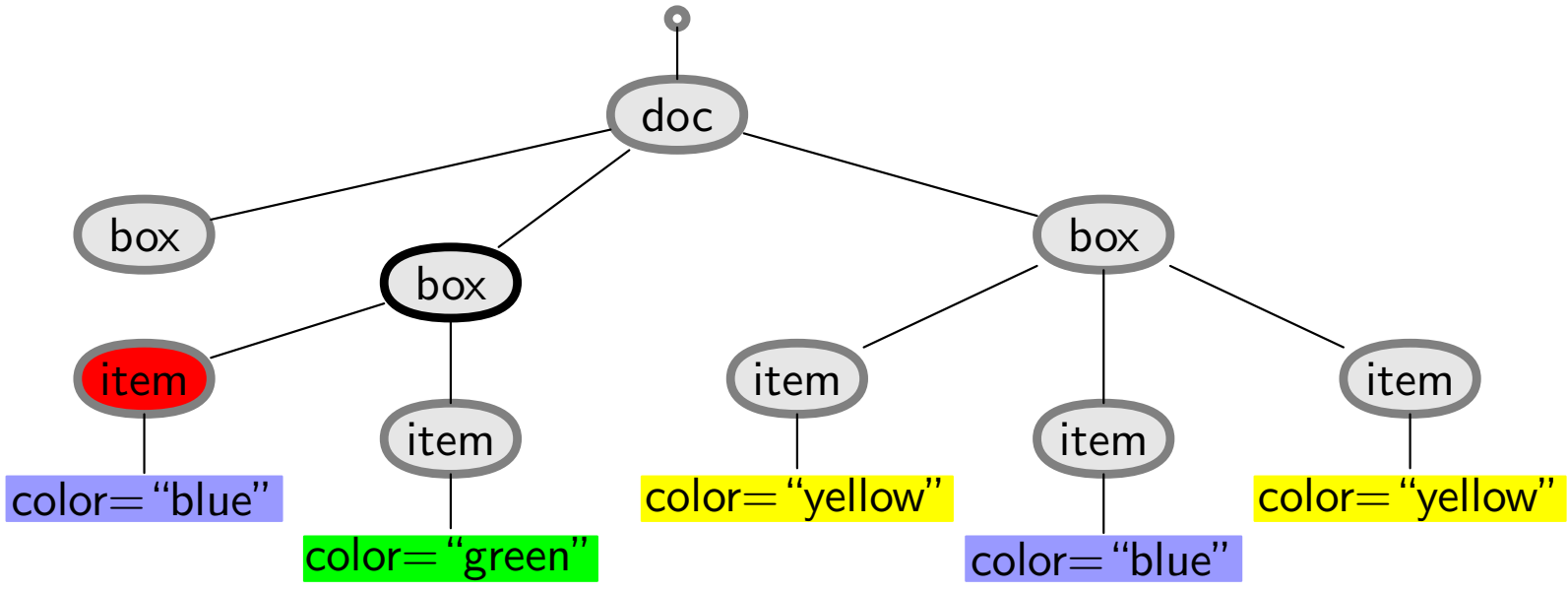
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

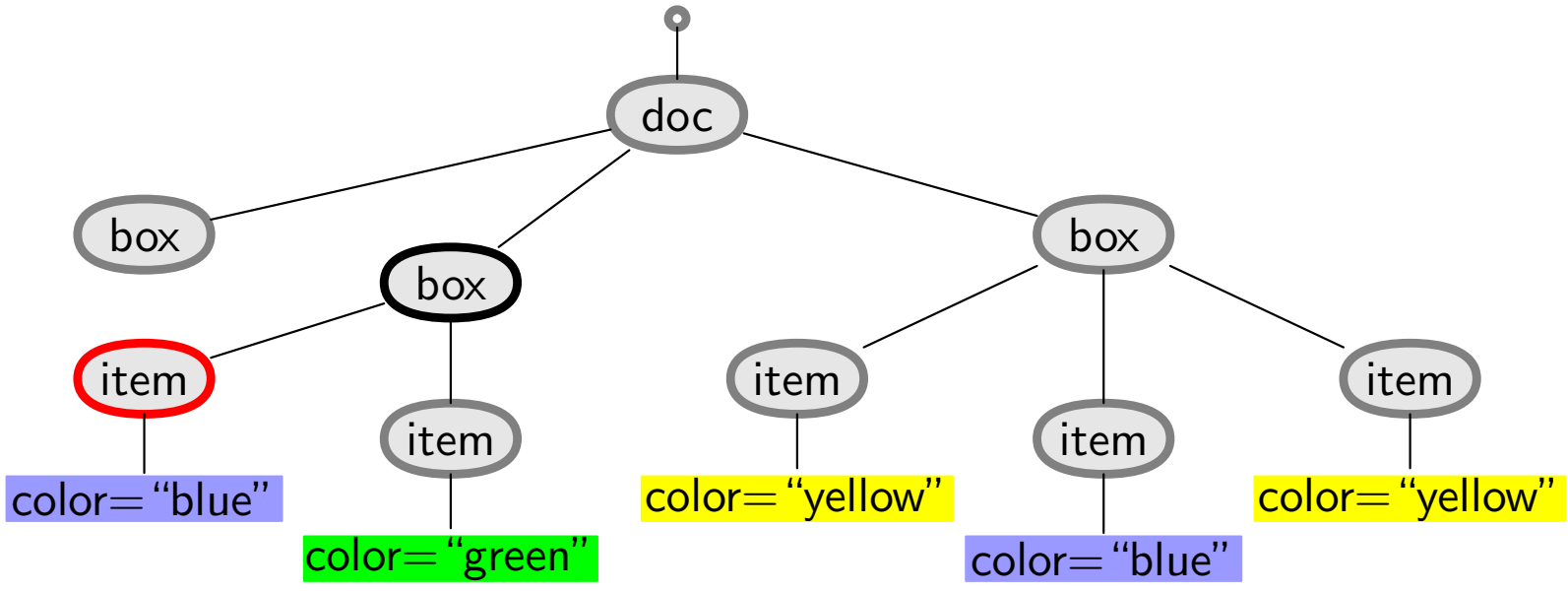
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

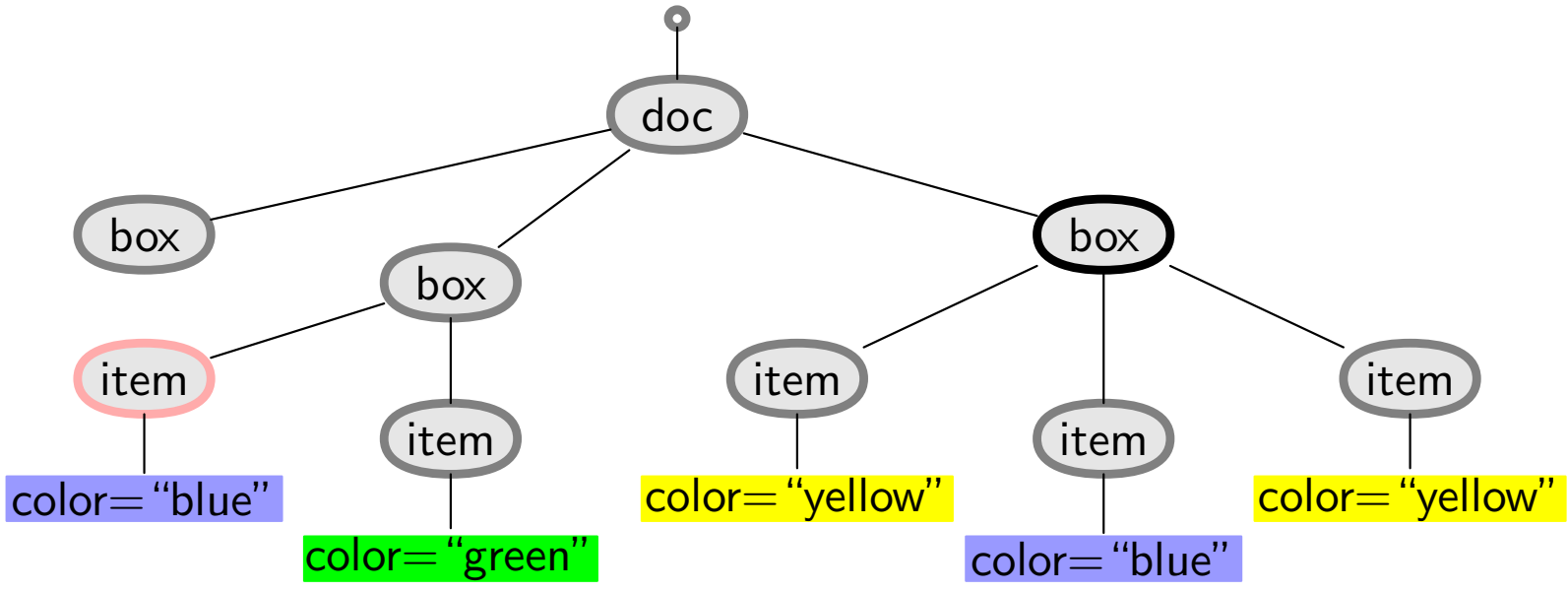
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

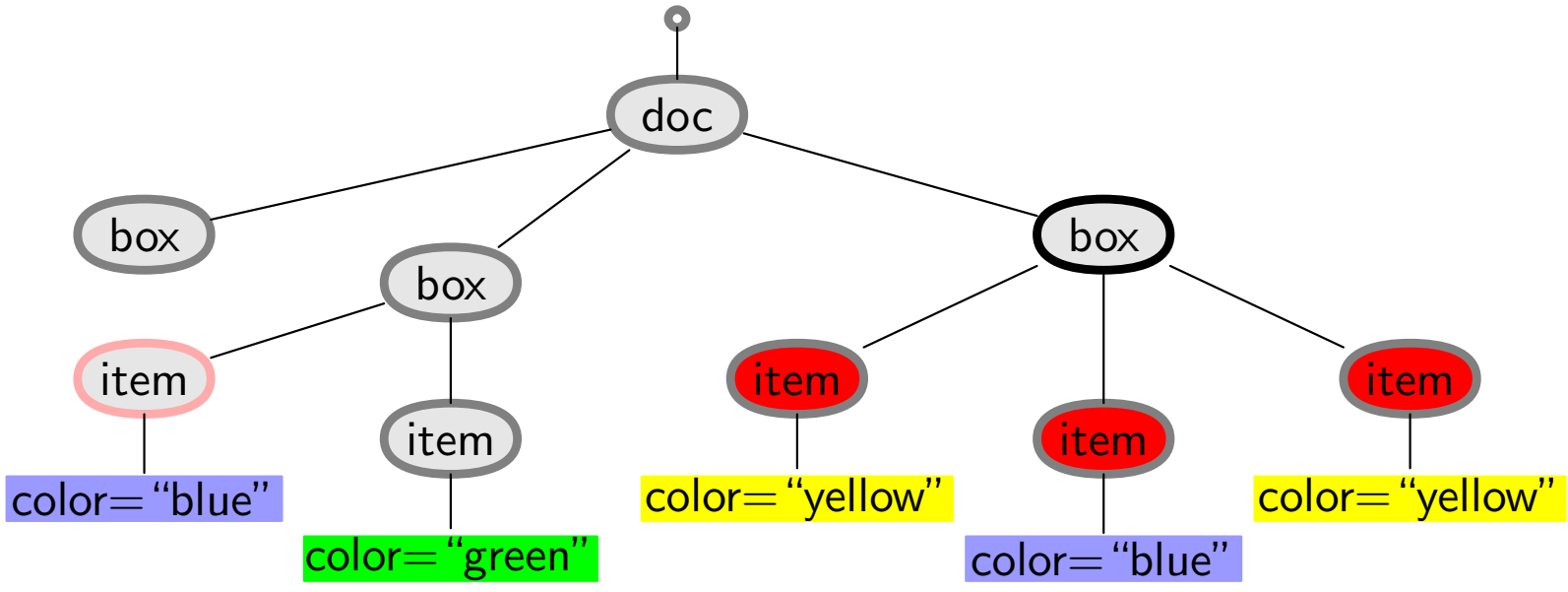
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

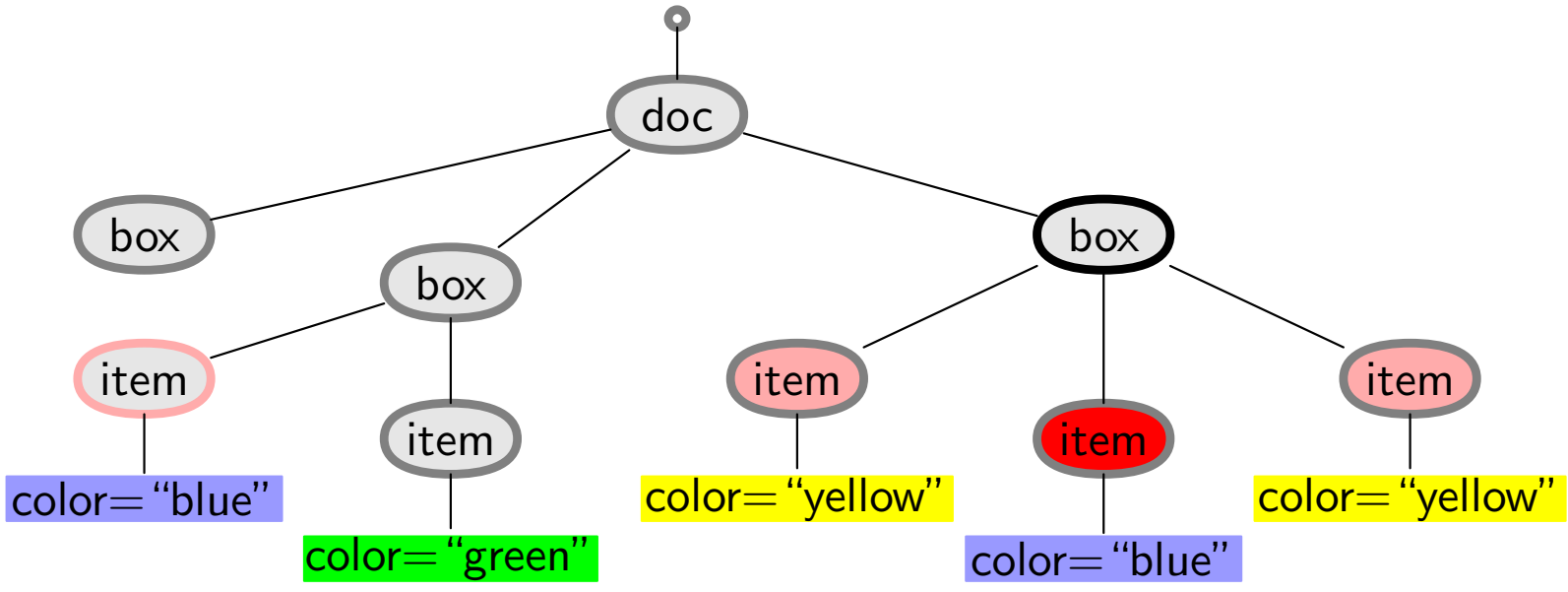
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

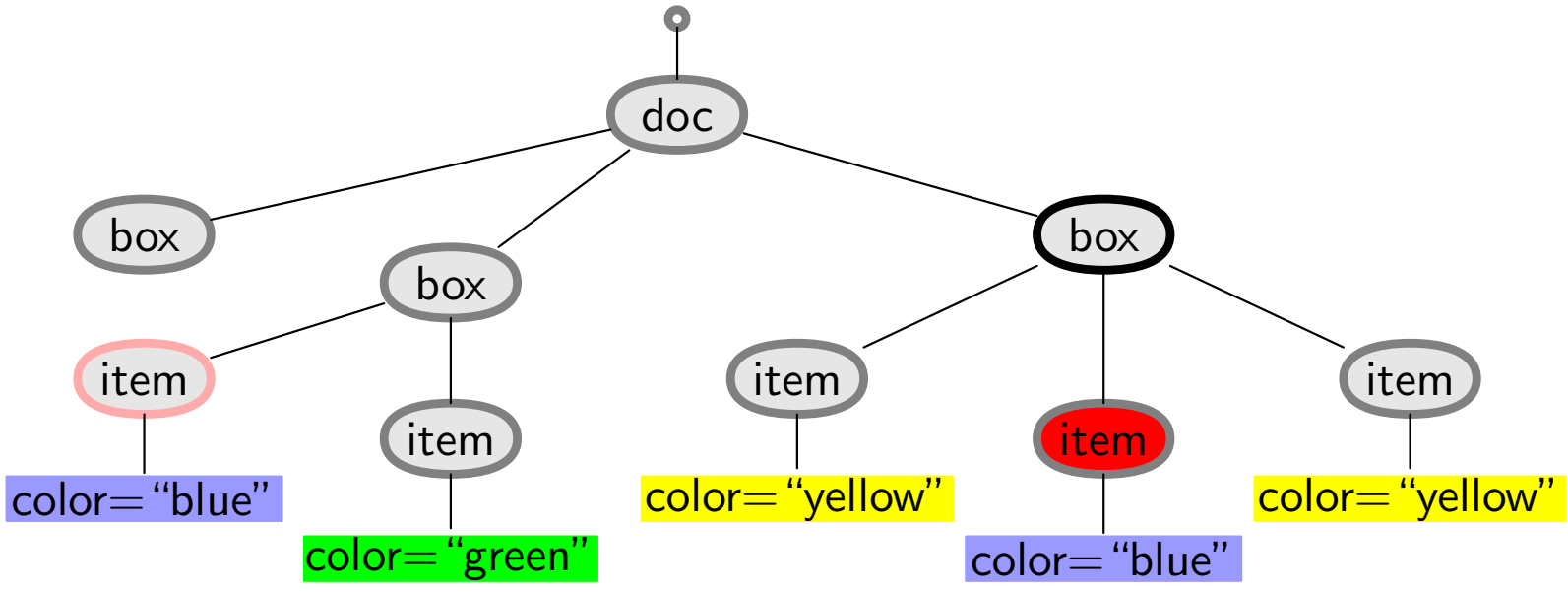
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

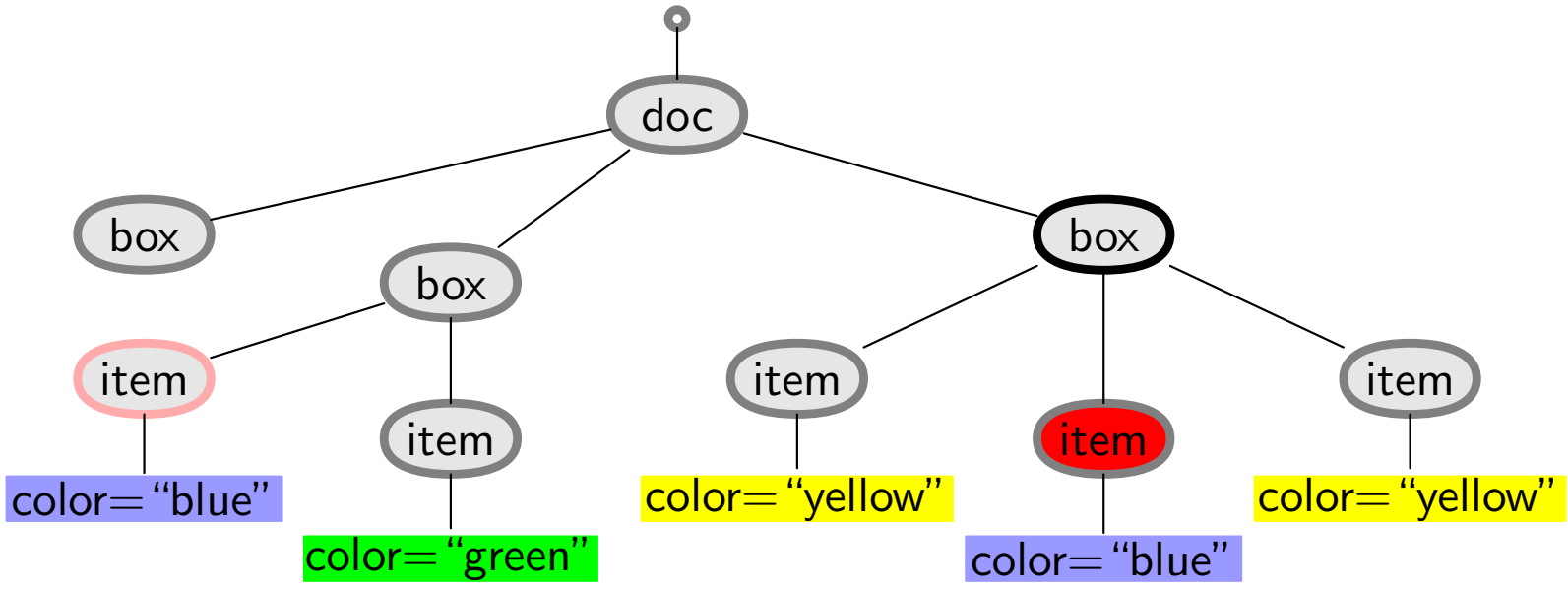
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

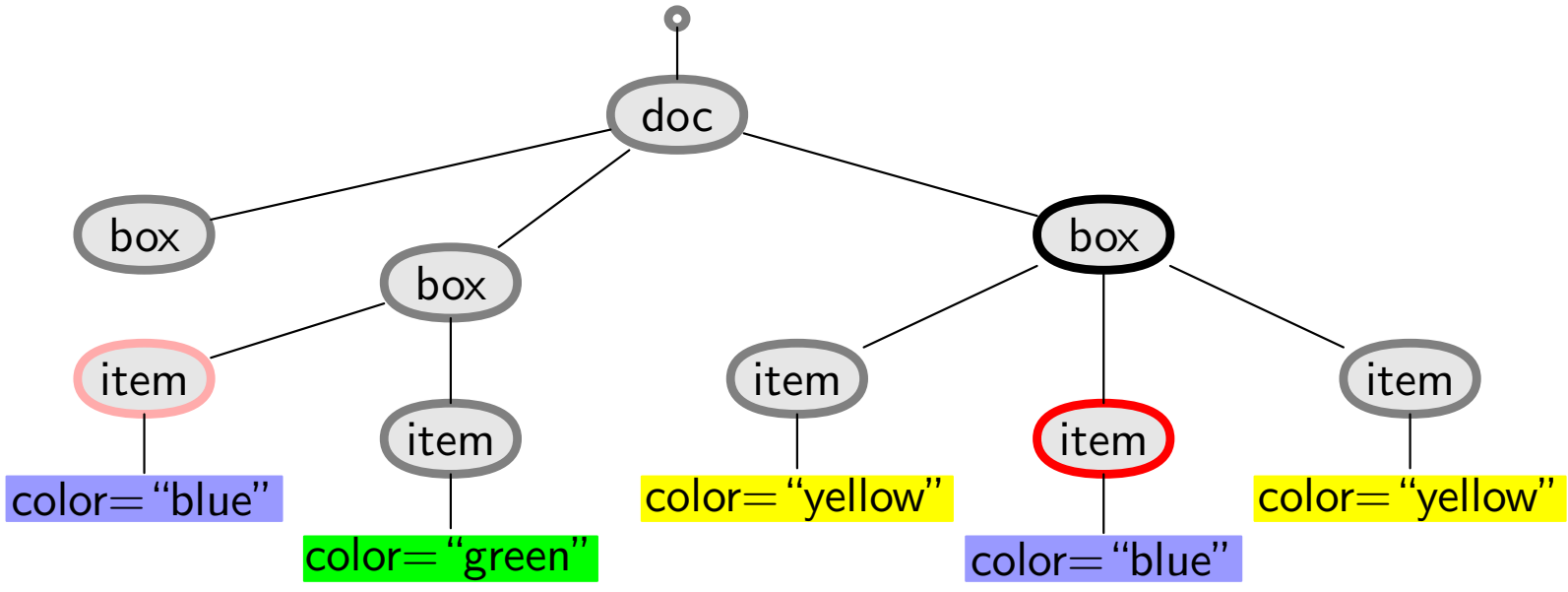
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

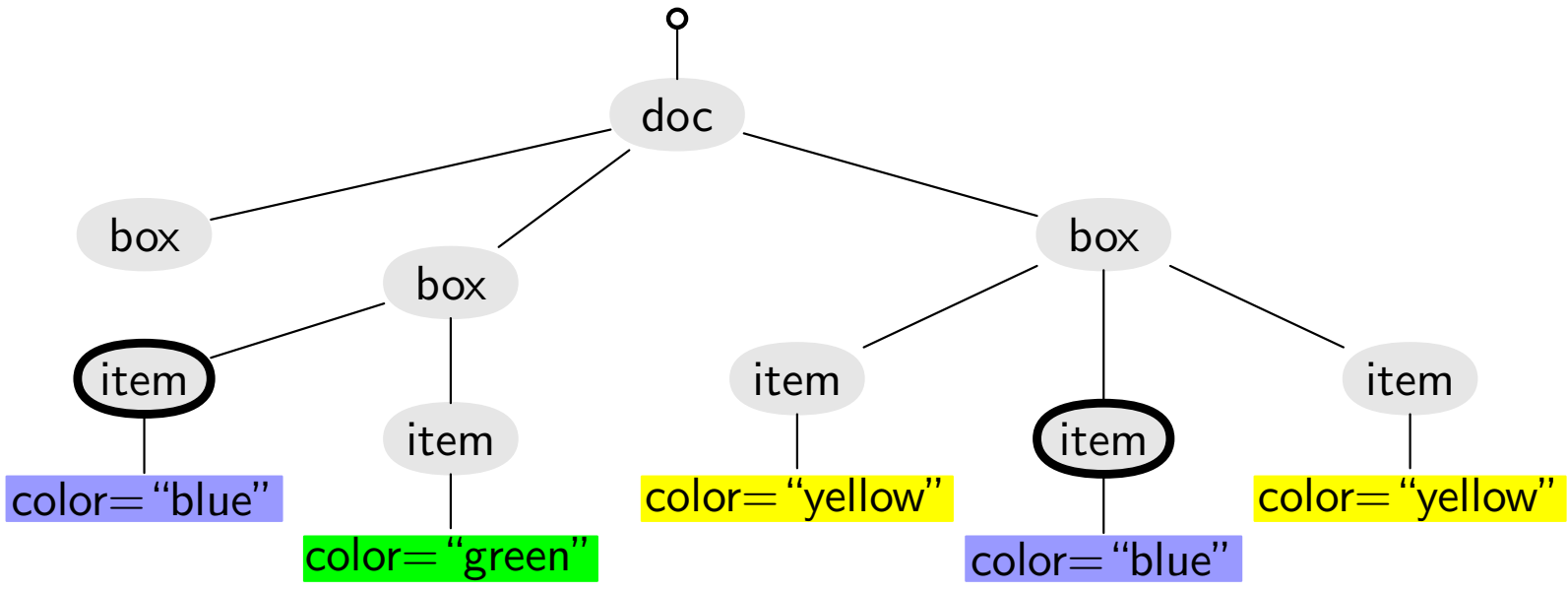
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

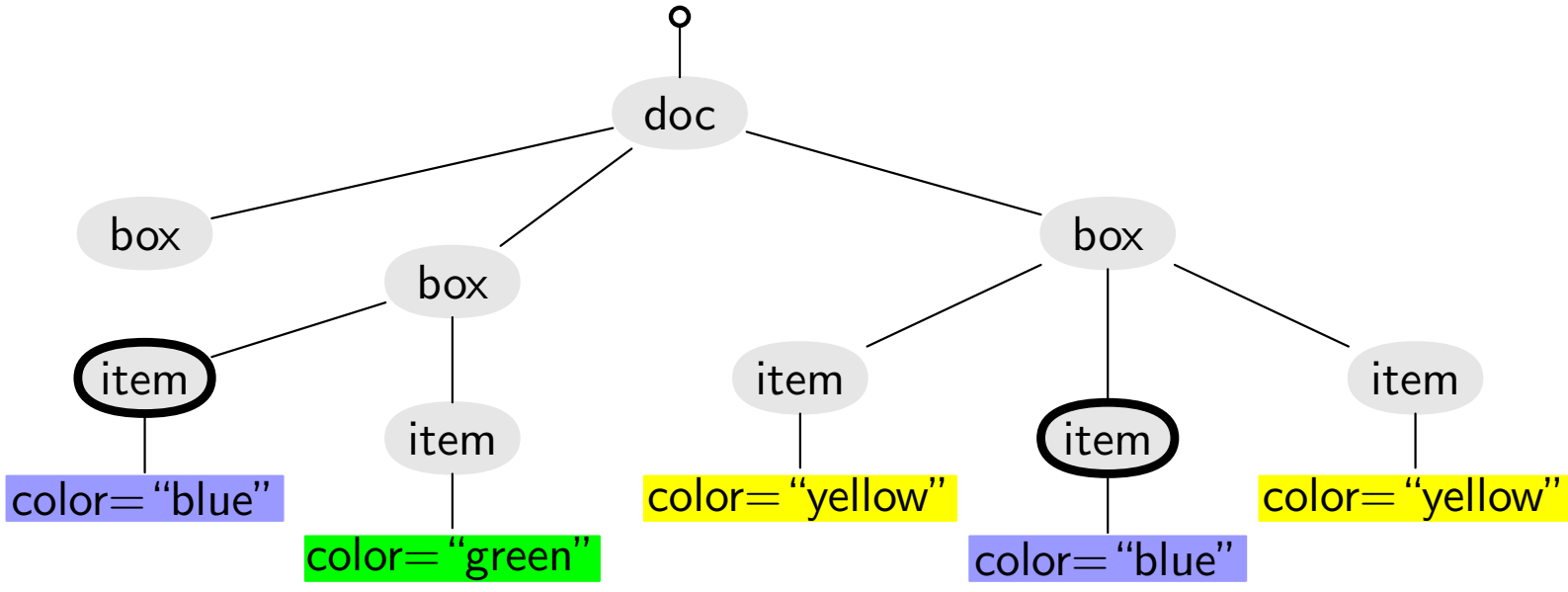
Document:



Query: `//item[@color="blue"][1]`

Commutation of Filter Expressions

Document:



Query: `//item[@color="blue"][1]`

Note the difference to `//item[1][@color="blue"]` !

Literature


A. Møller and M.I. Schwartzbach.
An Introduction to XML and Web Technologies.
Addison-Wesley, 2006.


M. Benedikt and C. Koch.
XPath leashed.
ACM Computing Surveys, to appear.


G. Gottlob, C. Koch, and R. Pichler.
Efficient algorithms for processing XPath queries.
ACM Transactions on Database Systems, 30:444–491, 2005.


P.T. Wood.
Minimising simple XPath expressions.
Workshop on the Web and Databases, 2001.


D. Olteanu, H. Meuss, T. Furche, and F. Bry.
XPath: Looking forward.
Workshop on XML Data Management, 2002.