The flagderiv package*

Paul van Tilburg
paul@luon.net

August 26, 2005

Contents

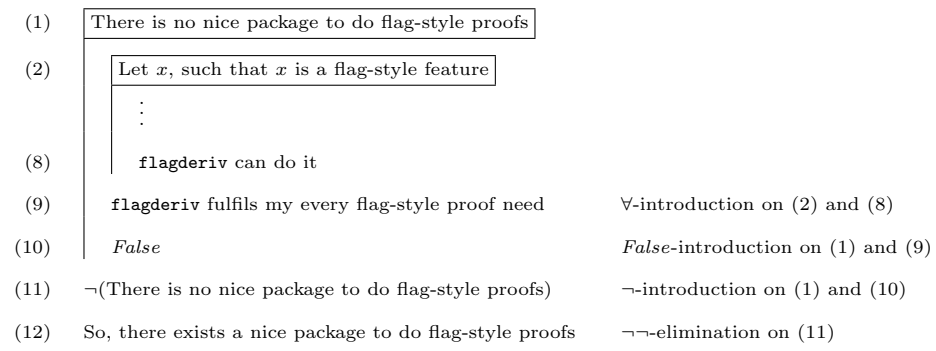
*This document corresponds to flagderiv v0.10, dated 2005/08/26.

1 Introduction

This document describes how to use `flagderiv` to make flag-style proofs in LaTeX.

It can handle taking steps, making assumptions, introducing variables and conclusions. The package allows the style to be configured. For example, the variable introduction symbol can be redefined. Other features include multiple page flag-style proofs.

The documentation and all examples are based upon version v0.10 of `flagderiv`.



2 Acknowledgements

I want to thank Jan Zwanenburg and Erik Poll for the first versions of this package, professor R.P. Nederpelt for his ever-persisting package reviews, feedback and suggestions and last but not least Mark van Eijk, who has written parts of section 3 and all examples in section 4.

3 Usage

This section explains how the `flagderiv` package works by means of examples and how the commands should be used. I try to cover all possibilities and features of this package. For bigger examples, please refer to section 4; for more detailed documentation on the commands and inner workings, refer to the `flagderiv` package itself.

3.1 Basic derivations

`flagderiv`

Basic derivations are written using an intuitive command sequence placed in the `flagderiv` environment. The example below uses auto-numbering of the lines. Variable introductions, assumptions, steps and conclusions can be done using the following commands respectively:

\introduce
- `\introduce{`⟨*label*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}`,

\assume
- `\assume{`⟨*label*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}`,

\step
- `\step{`⟨*label*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}`, and

\conclude
- `\conclude[`⟨*number*⟩`]{`⟨*label*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}`.

Lines can be referred to using the *label* that is provided as the first argument to each of the commands. *Formulas* are placed in the second argument and a *comment* in the third. See subsection 3.2 for the explanation of the optional argument *number* of `\conclude`.

Auto-numbering

```
\begin{flagderiv}
  \introduce{in-x}{x: \Nat}{Introduction of $x$}
    \assume{as-x}{x > 5}{Assumption}
      \step{big-x}{x > 1}{Arithmetic on \ref{in-x} and \ref{as-x}}
    \conclude{conc}{x > 5 \implies x > 1}
                   {$\implies$-intro on \ref{as-x} and \ref{big-x}}
  \conclude{}{\forall x \in \Nat: x > 5 \implies x > 1}
             {$\forall$-intro on \ref{in-x} and \ref{conc}}
\end{flagderiv}
```

(1)	**var** $x : \mathbb{N}$	Introduction of x
(2)	$x > 5$	Assumption
(3)	$x > 1$	Arithmetic on (1) and (2)
(4)	$x > 5 \Rightarrow x > 1$	$\Rightarrow$-intro on (2) and (3)
(5)	$\forall x \in \mathbb{N} : x > 5 \Rightarrow x > 1$	$\forall$-intro on (1) and (4)

\introduce*
\assume*
\step*
\conclude*

The following example uses manual numbering. This is done by using the starred versions of the previous given commands (`\introduce*`, `\assume*`, `\step*` and `\conclude*`) and using a *custom number* instead of a em label as the first argument. Note that because of this, derivation line references are not possible and should be done manually as well.

Manual numbering

```
\begin{flagderiv}
  \introduce*{$(1)$}{x: \Nat}{Introduction of $x$}
    \assume*{$(2)$}{P(x)}{Assumption}
      \skipsteps*{\dots}{}
      \step*{$(n - 2)$}{Q(x)}{}
    \conclude*{$(n - 1)$}{P(x) \implies Q(x)}
                         {$\implies$-intro on $(2)$ and $(n - 2)$}
  \conclude*{$(n)$}{\forall x \in \Nat: P(x) \implies Q(x)}
                   {$\forall$-intro on $(1)$ and $(n - 1)$}
\end{flagderiv}
```

(1)	**var** $x : \mathbb{N}$	Introduction of x
(2)	$P(x)$	Assumption
	...	
$(n-2)$	$Q(x)$	
$(n-1)$	$P(x) \Rightarrow Q(x)$	$\Rightarrow$-intro on (2) and $(n-2)$
(n)	$\forall x \in \mathbb{N} : P(x) \Rightarrow Q(x)$	$\forall$-intro on (1) and $(n-1)$

Note that the `\skipsteps*` used in the previous example is explained in section 3.3.

3.2 Concluding multiple assumptions/introductions

Sometimes it is useful to make a conclusion from multiple open assumptions and/or introductions at once. This can be done by supplying an optional *number* to the `\conclude` command. Usage is illustrated in the example below.

Multiple conclusion

```
\begin{flagderiv}
  \introduce{}{x: \Nat}{}
    \assume{}{P(x)}{}
      \step{}{Q(x)}{}
  \conclude[2]{}{\forall x \in \Nat: (P(x) \implies Q(x))}{}
\end{flagderiv}
```

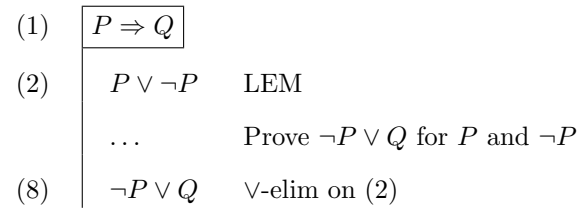
(1)	**var** $x : \mathbb{N}$
(2)	$P(x)$
(3)	$Q(x)$
(4)	$\forall x \in \mathbb{N} : (P(x) \Rightarrow Q(x))$

3.3 Skipping steps

`\skipsteps` When writing example derivations or abbreviating those where obvious/trivial steps may be skipped, one can use the `\skipsteps{`⟨*number*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}` command for auto-numbered derivations. This commands takes the *number* of steps to skip as the first argument, a *formula* as the second and a *comment* as the third, like a normal step.

Skipping 5 steps

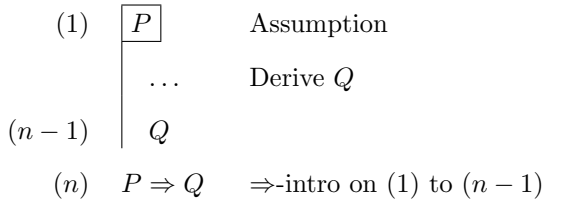
```
\begin{flagderiv}
  \assume{as}{P \implies Q}{}
    \step{lem-p}{P \lor \neg P}{LEM}
    \skipsteps{5}{\dots}
                 {Prove $\neg P \lor Q$ for $P$ and $\neg P$}
    \step{or-elim}{\neg P \lor Q}{$\lor$-elim on \ref{lem-p}}
\end{flagderiv}
```



`\skipsteps*` For skipping an unknown number of steps in a manual numbered derivation, the command `\skipsteps*{`⟨*formula*⟩`}{`⟨*comment*⟩`}` can be used. Here the number of steps to skip must be omitted, since there is no counting involved. See for example the ⇒-intro rule:

Skipping steps

```
\begin{flagderiv}
  \assume*{$(1)$}{P}{Assumption}
    \skipsteps*{\dots}{Derive $Q$}
    \step*{$(n - 1)$}{Q}{}
  \conclude*{$(n)$}{P \implies Q}
                   {$\implies$-intro on $(1)$ to $(n - 1)$}
\end{flagderiv}
```



3.4 Closing assumptions/introductions

`\done` Assumptions and introductions (contexts) are not always closed by a conclusion, it is sometimes useful to keep proven propositions within its context. The `\done` command (`\done[`⟨*number*⟩`]`) can be used to close a assumption and/or introduction without mentioning why. It takes an optional argument *number* to close multiple open assumptions and/or introductions. An example from the WTT (Weak Type Theory):

Closing assumptions/introductions

```
\begin{flagderiv}
  \introduce{}{n: \Nat}{}
    \introduce{}{d: \Nat}{}
      \step{}{\text{$d$ is a divisor of $n$} :=
               \exists_{k: \Nat} (n = d \cdot k)}{}
    \done
    \introduce{}{m: \Nat}{}
      \step{}{\text{$m$ is a multiple of $n$} :=
               \exists_{k: \Nat} (m = k \cdot n)}{}
  \done[2]
\end{flagderiv}
```

(1) **var** $n : \mathbb{N}$
(2) | **var** $d : \mathbb{N}$
(3) | d is a divisor of $n := \exists_{k:\mathbb{N}}(n = d \cdot k)$
(4) | **var** $m : \mathbb{N}$
(5) | m is a multiple of $n := \exists_{k:\mathbb{N}}(m = k \cdot n)$

The last `\done[2]` can be omitted since closing the `flagderiv` environment implicitly closes all open introductions and assumptions.

3.5 Spacing between lines

`derivskip` The spacing between derivation lines defaults to `8pt` but can be changed by adjusting the length (using `\setlength` on `derivskip`) before starting or during a `flagderiv` environment. See for example the more condensed proof:

Condensing a derivation

```
\setlength{\derivskip}{4pt}
\begin{flagderiv}
  \assume{}{P}{}
    \assume{}{Q}{}
      \step{}{P}{}
    \conclude{}{Q \implies P}{}
  \conclude{}{P \implies Q \implies P}{}
\end{flagderiv}
```

(1) P
(2) | Q
(3) | | P
(4) | $Q \Rightarrow P$
(5) $P \Rightarrow Q \Rightarrow P$

Note that flag derivations will automatically be split across pages if it is too long and the `flagderiv` environment is used. See also the `longtable` package. This does not happen for `flagderiv*`, which is a normal `tabular`.

3.6 The introduction symbol

`\introsymb` The way an introduction flag differs from an assumption flag is that it is prefixed with an *introduction symbol*. This symbol is by default set to **var** (`\mathbf{var}`) but can be changed to a different symbol, for example to $\rightsquigarrow$ (`\leadsto`) or just 'Let'. This can be changed by redefining the `\introsymb` command before opening the `flagderiv` environment.

Changing the introduction symbol

```
\renewcommand{\introsymb}{\textbf{Let}}
\begin{flagderiv}
  \introduce{intro-x}{x: \Nat}{Introduction}
    \introduce{intro-y}{y: \Nat}{Introduction}
      \skipsteps*{\ddots}{}
\end{flagderiv}
```

(1) **Let** $x : \mathbb{N}$ Introduction

(2) **Let** $y : \mathbb{N}$ Introduction

$\ddots$

3.7 Numbering and format

`fd@stepcount` `\thestepcount` When the derivation lines are automatically numbered the numbers are formatted as "(*number*)" and so are the references. This format can be changed by overriding the `\thestepcount` command using the `fd@stepcount` counter before opening the `flagderiv` environment.

Changing the numbering format

```
\renewcommand{\thestepcount}{[\Roman{fd@stepcount}]}
\begin{flagderiv}
  \step{ass}{P \implies Q}{}
  \step{lem}{P \lor \neg P}{LEM}
  \assume{ass-nP}{\neg P}{Assumption}
    \step{s0}{\neg P \lor \neg Q}{$\lor$-intro on \ref{ass-nP}}
  \done
  \assume{ass-nnP}{P}{Assumption}
    \step{s1}{Q}{$\implies$-elim on \ref{ass-nnP} and \ref{ass}}
    \step{s2}{\neg P \lor \neg Q}{$\lor$-intro on \ref{s1}}
  \done
  \step{or-el}{\neg P \lor Q}
              {$\lor$-elim on \ref{lem}, \ref{s0} and \ref{s2}}
\end{flagderiv}
```

[I]	$P \Rightarrow Q$	
[II]	$P \vee \neg P$	LEM
[III]	$\boxed{\neg P}$	Assumption
[IV]	$\neg P \vee \neg Q$	$\vee$-intro on [III]
[V]	$\boxed{P}$	Assumption
[VI]	Q	$\Rightarrow$-elim on [V] and [I]
[VII]	$\neg P \vee \neg Q$	$\vee$-intro on [VI]
[VIII]	$\neg P \vee Q$	$\vee$-elim on [II], [IV] and [VII]

`\thesteplabel` Since the manual numbered derivations use no counter, the format can be changed by overriding the `\thesteplabel` command which will take one argument, the *custom number*.

Changing the numbering format

```
\renewcommand{\thesteplabel}[1]{#1.}
\begin{flagderiv}
  \assume*{1}{P}{Assumption}
    \skipsteps*{\dots}{Derive $Q$}
    \step*{n - 1}{Q}{}
  \conclude*{n}{P \implies Q}{$\implies$-intro on 1 to n - 1}
\end{flagderiv}
```

1.	$\boxed{P}$	Assumption
	$\ldots$	Derive Q
n - 1.	Q	
n.	$P \Rightarrow Q$	$\Rightarrow$-intro on 1 to n - 1

Note that since references are also manually done, the same format should be used there by the writer him/herself.

3.8 Inline comments

In general there are two ways to place the comments with respect to the formulas in derivations. The first is behind the formula (the default). The second way is to place them inline, that is, on a separate line before the formula.

All `flagderiv` environments can be set globally to display the comments inline by passing the global option `inlcmnts` to the `flagderiv` package (e.g. `\usepackage[inlcmnts]{flagsderiv}`).

`\inlcmnts` `\noinlcmnts`

Giving the `\inlcmnts` command makes the environments after this command explicitly switch into inline-comment-mode (the default or global option is forgotten) after which giving `\noinlcmnts` will set it back to normal mode.

Inline comments

```
\inlcmnts
\begin{flagderiv}
  \assume{as-P}{P}{Assume:}
    \assume{as-Q}{Q}{Assume:}
      \step{re-P}{P}{Rei \ref{as-P}:}
   \conclude{imp1}{Q \implies P}
                  {$\implies$-intro on \ref{as-Q} and \ref{re-P}:}
  \conclude{imp2}{P \implies Q \implies P}
                  {$\implies$-intro on \ref{as-P} and \ref{imp1}:}
\end{flagderiv}
```

{ Assume: }

(1) $\boxed{P}$

  { Assume: }

(2) $\boxed{Q}$

    { Rei (1): }

(3) P

  { $\Rightarrow$-intro on (2) and (3): }

(4) $Q \Rightarrow P$

{ $\Rightarrow$-intro on (1) and (4): }

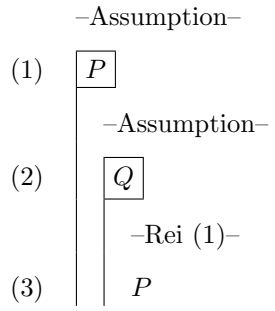
(5) $P \Rightarrow Q \Rightarrow P$

`\theinlcmnt`

When the comments are inserted inline, a certain format is used, namely to enclose the comment with braces. This is the default, but can be overridden by redefining the `\theinlcmnt` command.

Changing inline comments format

```
\inlcmnts
\renewcommand{\theinlcmnt}[1]{--#1--}
\begin{flagderiv}
  \assume{ass-P}{P}{Assumption}
    \assume{ass-Q}{Q}{Assumption}
      \step{rei-P}{P}{Rei \ref{ass-P}}
\end{flagderiv}
```



3.9 Namespaces for labels

In a document with a lot of auto-numbered derivations it is easy to lose overview of labels already used or it may be hard to keep thinking of new ones. To solve this a `flagderiv` environment can take a *namespace* as optional argument. This makes the lines referable as normal within the same derivation but also globally when the namespace and a colon is added outside the derivation. For example a derivation for Modus Tollens (MT):

Derivation label namespace

```
\begin{flagderiv}[mt]
  \step{s0}{P \implies Q}{}
  \assume{as-nQ}{\neg Q}{Assumption}
    \assume{as-P}{P}{Assumption}
      \step{s1}{Q}{$\implies$-elim on \ref{s0} and \ref{as-P}}
      \step{s2}{\bot}{$\bot$-intro on \ref{as-nQ} and \ref{s1}}
    \conclude{nP}{\neg P}{$\neg$-intro on \ref{as-P} and \ref{s2}}
  \conclude{nQ-nP}{\neg Q \implies \neg P}
                  {$\implies$-intro on \ref{as-nQ} and \ref{nP}}
\end{flagderiv}

Note that step~\ref{mt:as-P} is not an intuitive step but necessary
to get $\neg P$ under the assumption $\neg Q$.
```

(1)	$P \Rightarrow Q$	
(2)	$\boxed{\neg Q}$	Assumption
(3)	$\quad\boxed{P}$	Assumption
(4)	$\quad\quad Q$	$\Rightarrow$-elim on (1) and (3)
(5)	$\quad\quad \bot$	$\bot$-intro on (2) and (4)
(6)	$\quad \neg P$	$\neg$-intro on (3) and (5)
(7)	$\neg Q \Rightarrow \neg P$	$\Rightarrow$-intro on (2) and (6)

Note that step (3) is not an intuitive step but necessary to get $\neg P$ under the assumption $\neg Q$.

3.10 Other options

\caption
\tablehead
\tablefirsthead
\endfoot
\endlastfoot
flagderiv*

A `flagderiv` environment actually wraps a `longtable`. This means that at the start of an environment all `longtable` options are available. This means the use of `\caption`, and setting heads (`\tablehead` and `\tablefirsthead`) and feet (`\endfoot` and `\endlastfoot`) but also changing the lengths, etc.

Note that this does not hold for the slightly different `flagderiv*` environment, which acts exactly the same but is wrapped in a normal `tabular` and thus can not be split across pages.

Captioned

```
\inlcmnts
\begin{flagderiv}[reflex]
  \caption{The reflexivity of $\equiv_n$} \\
  \introduce*{(1)}{x \in \All}{Assume:}
    \skipsteps*{\dots \\ \dots}{}
    \step*{(n - 1)}{x \equiv_n x}{}
  \conclude*{(n)}{\forall x \in \All: x \equiv_n x}
                 {$\forall$-intro on (1) and (n - 1):}
\end{flagderiv}
```

Derivation 15: The reflexivity of $\equiv_n$

	{ Assume: }
(1)	**var** $x \in \mathbb{Z}$
	...
	...
(n - 1)	$x \equiv_n x$
	{ $\forall$-intro on (1) and (n - 1): }
(n)	$\forall x \in \mathbb{Z} : x \equiv_n x$

\\ Additional it is possible to add custom newlines (\\) in all step formulas[1] when the formula gets too long as demonstrated in the above example.

[1] This is not possible in introductions or assumptions, but they can often be split into multiple introductions or assumptions.

4 Examples

The following are just a few real-life examples of flag-style proofs, where `flagderiv` provides an elegant means to include these proofs in LaTeX-documents.

4.1 Examples from `flagderiv`

A derivation with auto-numbering and labels/references

(1)	$\boxed{\textbf{var } x, y \in \mathbb{N}}$	Introduction
(2)	$\quad\boxed{x > 0}$	Assumption
(3)	$\quad\quad\boxed{x < y}$	Assumption
(4)	$\quad\quad y > 0$	Transitivity on (2) and (3)
(5)	$\quad\quad y^3 > 0$	Arithmetic on (4)
(6)	$\quad x < y \Rightarrow y^3 > 0$	$\Rightarrow$-intro on (3) and (5)
(7)	$x > 0 \Rightarrow x < y \Rightarrow y^3 > 0$	$\Rightarrow$-intro on (2) and (6)
(8)	$\forall x, y \in \mathbb{N} : x > 0 \wedge x < y \Rightarrow y^3 > 0$	$\forall$-intro on (1) and (7)

A derivation with manual numbering

(m)	$\boxed{\neg P}$	
	$\quad\ldots$	
(n - 2)	$\quad False$	
(n - 1)	$\neg\neg P$	$\Rightarrow$-intro on (m) and (n - 2), and Negation
(n)	P	Double negation on (n - 1)

4.2 Predicate Calculus

Idempotence

(1)	$\boxed{P \wedge P}$	
(2)	$\quad P$	$\wedge$-elim on (1)
(3)	$P \wedge P \Rightarrow P$	$\Rightarrow$-intro on (1) and (2)
(4)	$\boxed{P}$	

(5)	$P \wedge P$	$\wedge$-intro on (4) and (4)
(6)	$P \Rightarrow P \wedge P$	$\Rightarrow$-intro on (4) and (5)
(7)	$(P \wedge P \Rightarrow P) \wedge (P \Rightarrow P \wedge P)$	$\wedge$-intro on (3) and (6)
(8)	$P \wedge P \equiv P$	$\equiv$-intro on (7)

De Morgan

(1)	$\boxed{\neg(P \wedge Q)}$	
(2)	$\quad\boxed{\neg\neg P}$	
(3)	$\quad\quad\boxed{Q}$	
(4)	$\quad\quad P$	$\neg\neg$-elim on (2)
(5)	$\quad\quad P \wedge Q$	$\wedge$-intro on (3) and (4)
(6)	$\quad\quad \mathit{False}$	*False*-intro on (5) and (1)
(7)	$\quad\neg Q$	$\neg$-intro on (3) and (6)
(8)	$\neg\neg P \Rightarrow \neg Q$	$\Rightarrow$-intro on (2) and (7)
(9)	$\neg P \vee \neg Q$	$\vee$-intro on (8)
(10)	$\neg(P \wedge Q) \Rightarrow \neg P \vee \neg Q$	$\Rightarrow$-intro on (1) and (9)
(11)	$\boxed{\neg P \wedge \neg Q}$	
(12)	$\quad\boxed{P \vee Q}$	
(13)	$\quad\quad \neg P \Rightarrow Q$	$\vee$-elim on (12)
(14)	$\quad\quad \neg P$	$\wedge$-elim on (11)
(15)	$\quad\quad Q$	$\Rightarrow$-elim on (13) and (14)
(16)	$\quad\quad \neg Q$	$\wedge$-elim on (11)
(17)	$\quad\quad \mathit{False}$	*False*-intro on (15) and (16)
(18)	$\quad P \vee Q \Rightarrow \mathit{False}$	$\Rightarrow$-intro on (12) and (17)
(19)	$\quad \neg(P \vee Q)$	$\neg$-intro on (18)
(20)	$\neg P \wedge \neg Q \Rightarrow \neg(P \vee Q)$	$\Rightarrow$-intro on (11) and (19)
(21)	$(\neg(P \wedge Q) \Rightarrow \neg P \vee \neg Q) \wedge$ $(\neg P \wedge \neg Q \Rightarrow \neg(P \vee Q))$	$\wedge$-intro on (10) and (20)
(22)	$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$	$\equiv$-intro on (21)

Transitivity

(1)	**var** P, Q, R	
(2)	$(P \Rightarrow Q) \wedge (Q \Rightarrow R)$	
(3)	P	
(4)	$P \Rightarrow Q$	$\wedge$-elim on (2)
(5)	Q	$\Rightarrow$-elim on (4) and (3)
(6)	$Q \Rightarrow R$	$\wedge$-elim on (2)
(7)	R	$\Rightarrow$-elim on (6) and (5)
(8)	$P \Rightarrow R$	$\Rightarrow$-intro on (3) and (7)
(9)	$(P \Rightarrow Q) \wedge (Q \Rightarrow R) \Rightarrow (P \Rightarrow R)$	$\Rightarrow$-intro on (2) and (8)
(10)	$(\forall P, Q, R :: (P \Rightarrow Q) \wedge (Q \Rightarrow R) \Rightarrow (P \Rightarrow R))$	$\forall$-intro on (1) and (9)

4.3 Rewrite Systems

Mixed confluent $\Leftrightarrow$ Church-Rosser

1	*CR*	hyp
2	Let $a, b, c \in A,$ such that $a \rightarrow b \wedge a \twoheadrightarrow c$	
3	$a \rightarrow b \wedge a \twoheadrightarrow c$	rei (2)
4	$a \twoheadrightarrow b \wedge a \twoheadrightarrow c$	$\forall$-elimination on lemma 1 and (3)
5	$(\exists d \in A :: b \twoheadrightarrow d \wedge c \twoheadrightarrow d)$	$\forall$-elimination on (1) and (4)
6	*mixed confluent*	$\forall$-introduction on (2) and (5)
7	*CR* $\Rightarrow$ *mixed confluent*	$\Rightarrow$-introduction on (1) and (6)
8	*mixed confluent*	
9	Induction on the length of $a \twoheadrightarrow b$	
	Base :	
11	Let $a, b, c \in A,$ such that $a \xrightarrow{0} b \wedge a \twoheadrightarrow c$	
12	$a = b$	def $\xrightarrow{0}$, (11)
13	$a \twoheadrightarrow c$	$\wedge$-elimination on (11)
14	$b \twoheadrightarrow c$	(12) applied on (13)

15	$c \twoheadrightarrow c \wedge b \twoheadrightarrow c$	
16	$(\exists d \in A :: b \twoheadrightarrow d \wedge c \twoheadrightarrow d)$	
	Induction-hypothesis :	
18	$(\forall a, b, c \in A : a \xrightarrow{n} b \wedge a \twoheadrightarrow c :$	
	$(\exists d \in A :: b \twoheadrightarrow d \wedge c \twoheadrightarrow d))$	
	Step :	
20	Let $a, b, c \in A$, such that $a \xrightarrow{n+1} b \wedge a \twoheadrightarrow c$	
21	$a \xrightarrow{n+1} b$	$\wedge$-elimination on (20)
22	$a \xrightarrow{n} b' \rightarrow b$	
23	$a \twoheadrightarrow c$	$\wedge$-elimination on (20)
24	$a \xrightarrow{n} b' \wedge a \twoheadrightarrow c$	$\wedge$-introduction on (22) and (23)
25	$(\exists d' \in A :: b' \twoheadrightarrow d' \wedge c \twoheadrightarrow d')$	$\forall$-elimination on (18) and (24)
26	d', such that $b' \twoheadrightarrow d' \wedge c \twoheadrightarrow d'$	$\exists$-elimination on (25)
27	$b' \rightarrow b \wedge b' \twoheadrightarrow d'$	
28	$(\exists d \in A :: b \twoheadrightarrow d \wedge d' \twoheadrightarrow d)$	$\forall$-elimination on (8) and (27)
29	d, such that $b \twoheadrightarrow d \wedge d' \twoheadrightarrow d$	$\exists$-elimination on (28)
30	$a \xrightarrow{n} b' \rightarrow b \twoheadrightarrow d \wedge a \twoheadrightarrow c \twoheadrightarrow d' \twoheadrightarrow d$	
31	$b \twoheadrightarrow d \wedge c \twoheadrightarrow d$	
32	$(\exists d \in A :: b \twoheadrightarrow d \wedge c \twoheadrightarrow d)$	
33	*CR*	
34	*mixed confluent* $\Rightarrow$ *CR*	$\Rightarrow$-introduction on (8) and (33)
35	*mixed confluent* $\Leftrightarrow$ *CR*	$\Leftrightarrow$-introduction on (7) and (34)

Lemma 1 $(\forall a, b \in A : a \rightarrow b : a \twoheadrightarrow b)$

5 Implementation

First some initial code is needed for processing of the options. Note that `ifthen` is used throughout the package.

```
\RequirePackage{ifthen}
\newboolean{@inlcmnts}
```

Processing of the option (`inlcmnts`) for enabling inline comments.

```
\DeclareOption{inlcmnts}{\setboolean{@inlcmnts}{true}}
\ProcessOptions
```

Loading of the required packages. Note that `flagderiv` needs to require a fix/override for the array package, until the fixed version is available in most distributions.

```
\RequirePackage{array}
\RequirePackage{longtable}
\long\@namedef{NC@rewrite@*}#1#2{%
   \count@#1\relax
   \loop
   \ifnum\count@>\z@
   \advance\count@\m@ne
   \@temptokena\expandafter{\the\@temptokena#2}%
   \repeat
   \NC@find}
```

5.1 Generic settings, counters and commands

`fd@flagcount` `fd@stepcount` The counters keeping the number of opened/nested flags and the number of derivation steps.

```
\newcounter{fd@flagcount}
\newcounter{fd@stepcount}
```

`\derivskip` The default space (`8pt`) between lines in a derivation. This is overridable to define a different default length.

```
\newlength{\derivskip}
\setlength{\derivskip}{8pt}
```

`\introsymb` The symbol used as variable introduction prefix. Overridable with `\renewcommand` to insert a different symbol, defaults to: `\textbf{var}`.

Ex. `\renewcommand{\introsymb}{\textbf{Let}}`

```
\newcommand{\introsymb}{\textbf{var}}
```

`\thestepcount` This is the command used to generate a step number label (used by autonumbering). `\thestepcount` can be overridden to display labels differently, defaults to: (*number*). This exposes the internal `\thefd@stepcount` command.

Ex. `\renewcommand{\thestepcount}{[\arabic{stepcount}]}`

```
\newcommand{\thestepcount}{(\arabic{fd@stepcount})}
\renewcommand{\thefd@stepcount}{\thestepcount}
```

`\thesteplabel` Command used to generate a customized step label (used by manual numbering). `\thesteplabel` can be overridden to display manual numbers differently, this command has one argument: `{`⟨*label*⟩`}`. The command does nothing by default (i.e. just returns the argument).

Ex. `\renewcommand{\thesteplabel}[1]{[##1]}`

```
\newcommand{\thesteplabel}[1]{#1}
```

`\theinlcmnt` The command used to format inline comments (if enabled). `\theinlcmnt` can be overridden to display it differently and has one argument: `{`⟨*comment*⟩`}`. Defaults to: { *comment text* }.

Ex. `\renewcommand{\theinlcmnt}[1]{--#1--}`

```
\newcommand{\theinlcmnt}[1]{\{ #1 \}}
```

`\inlcmnts` Command to switch inline comments explicitly on (and by that, overriding and forgetting the default determined by the absence or presence of the `inlcmnts` package option).

```
\newcommand{\inlcmnts}{\setboolean{@inlcmnts}{true}}
```

`\noinlcmnts` Command to switch inline comments explicitly off (and by that, overriding and forgetting the default determined by the absence or presence of the `inlcmnts` package option).

```
\newcommand{\noinlcmnts}{\setboolean{@inlcmnts}{false}}
```

5.2 The derivation environment

The following three sections describe the internal commands and constructs of the `flagderiv` environment, after which the last section will explain how the environment works and how it is put together using the internal constructs.

Note that for almost all commands there exists two versions, one used for manual numbered derivations and one for automatic numbered/labelled derivations (both forms can be mixed within the environment).

5.2.1 Labels

Derivation lines are labelled and can be referred to. This can be done with automatically numbered lines where the first argument is a normal LaTeX label. However, to avoid clashing, the labels can be automatically prefixed with a namespace. The following two commands handle this prefixing.

`\@labelprefix` Command that specifies the label namespace prefix.

```
\newcommand{\@labelprefix}{\relax}
```

`\@derlabel` Internal command to define a label using the prefix (if a label is defined, otherwise do nothing).

```
\newcommand{\@derlabel}[1]{%
  \ifthenelse{\equal{#1}{}}{}{\label{\@labelprefix#1}}%
}
```

5.2.2 Steps

A line of a derivation can be one of five types:

1. A simple and bare line, only internally accessible,
2. a comment line,
3. a derivation step (both automatically and manual numbered),
4. a line to indicate skipping of steps,
5. a flag (handled in the next section).

This section deals with the first four types.

`\@derline` A simple derivation line that uses `fd@flagcount` to remember the number of open flags. This command is used as: `\@derline{`⟨*label*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}`, with:

`{`⟨***label***⟩`}` a label which can be left empty, to refer back to in comments of other derivation lines,

`{`⟨***formula***⟩`}` the formula, and

`{`⟨***comment***⟩`}` a comment.

```
\newcommand{\@derline}[3]{%
 \mbox{#1} &
 \setlength{\extrarowheight}{\derivskip}%
```

If the flag counter is still zero...

```
 \ifthenelse{\value{fd@flagcount}=0}%
```

then this line is outside any flag,

```
 {\begin{array}[t]{@{}l}}%
```

otherwise this is inside one or more flag contexts and the amount of open flagpoles equal to the number of open contexts should be inserted.

```
{\begin{array}[t]{*{\value{fd@flagcount}}{|@{\hspace{2\arraycolsep}}}l}}
 \ensuremath{#2}
\end{array} &
\mbox{#3} \\
}
```

`\@CMNTderline` Command to fold the comment on a separate line before the actual derivation line if the `inlcmnts` option is set for this package, see also `\@derline` for the argument handling.

```
\newcommand{\@CMNTderline}[3]{%
 \ifthenelse{\boolean{@inlcmnts}}{%
```

If the inline comments option is enabled, fold the comment before the actual line.

```
  \ifthenelse{\equal{#3}{}}{}{%
```

There is comment, insert it inline.

```
   \@derline{}{\mbox{\theinlcmnt{#3}}}{}%
  }
  \@derline{#1}{#2}{}%
 }{%
```

If inline comments are not enabled, pass everything to `\@derline` directly.

```
  \@derline{#1}{#2}{#3}%
 }%
}
```

`\@MANstep` `\@AUTOstep` Command for the manual (`MAN`) and automatically (`AUTO`) numbered version of a derivation step. See also `\@derline`.

```
\newcommand{\@MANstep}[3]{\@CMNTderline{\thesteplabel{#1}}{#2}{#3}}
```

```
\newcommand{\@AUTOstep}[3]{%
 \refstepcounter{fd@stepcount}%
 \@derlabel{#1}\@CMNTderline{\thestepcount}{#2}{#3}%
}
```

`\@MANskipsteps` `\@AUTOskipsteps` Command for the manual (`MAN`) and automatically (`AUTO`) numbered version of skipping derivation steps. See also `\@derline`.

```
\newcommand{\@MANskipsteps}[2]{\@CMNTderline{}{#1}{#2}}
```

```
\newcommand{\@AUTOskipsteps}[3]{%
 \addtocounter{fd@stepcount}{#1}%
 \@CMNTderline{}{#2}{#3}%
}
```

5.2.3 Flags

Flags are actually simple derivation lines with two extras: the formula is put into a flag box and the flag counter `fd@flagcount` is incremented when one is opened. Next to flag commands there are also commands to to close (and by that decrementing the counter) flags.

There exist three types of flags:

1. the simple and bare flag,
2. the assumption flag (both manual and automatically numbered),
3. the variable introduction flag (also manual and automatically numbered).

`\@flagbox` Creates a flag for (a box around) the given text/formula, takes `{`⟨*formula*⟩`}` as an argument.

```
\newcommand{\@flagbox}[1]{%
 \setlength{\fboxsep}{0.75ex}%
 \fbox{#1}%
}
```

`\@startflag` Starts a flag context by opening a flag and incrementing the counter. The command is used as: `\@startflag{`⟨*label*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}`, see also `\@derline` for the meaning of these arguments.

```
\newcommand{\@startflag}[3]{%
 \@CMNTderline{#1}{\@flagbox{\ensuremath{#2}}}{#3}%
 \addtocounter{fd@flagcount}{1}%
}
```

`\@endflag` Ends a flag by closing it and decrementing the counter. Takes the number of flags to close `{`⟨*number*⟩`}` as an argument.

```
\newcommand{\@endflag}[1]{\addtocounter{fd@flagcount}{-#1}}
```

`\@flagclose` Wrapper command for closing a number of flags simultaneously taking `[`⟨*number*⟩`]` as an optional argument, defaulting to 1. See also `\@endflag`.

```
\newcommand{\@flagclose}[1][1]{\@endflag{#1}}
```

`\@MANassume` `\@AUTOassume` Command for the manual (`MAN`) and automatic (`AUTO`) numbered version of an assumption flag, see also `\@startflag`.

```
\newcommand{\@MANassume}[3]{\@startflag{\thesteplabel{#1}}{#2}{#3}}
```

```
\newcommand{\@AUTOassume}[3]{
 \refstepcounter{fd@stepcount}%
 \@derlabel{#1}\@startflag{\thestepcount}{#2}{#3}%
}
```

\@MANintroduction
\@AUTOintroduction

Command for the manual (MAN) and automatically (AUTO) numbered version of an introduction flag, see also \@startflag.

```
\newcommand{\@MANintroduction}[3]{%
  \@startflag{\thesteplabel{#1}}{\introsymb~#2}{#3}%
}

\newcommand{\@AUTOintroduction}[3]{
  \refstepcounter{fd@stepcount}%
  \@derlabel{#1}\@startflag{\thestepcount}{\introsymb~#2}{#3}%
}
```

\@MANconclude
\@AUTOconclude

Command to conclude from one or more open flags in the manual (MAN) and automatically (AUTO) numbered version, see also \@endflag. The command is used as: \@MANconclude[⟨*number*⟩]{⟨*label*⟩}{⟨*formula*⟩}{⟨*comment*⟩}, where the [⟨*number*⟩] indicates the number of flags to close, defaulting to 1.

```
\newcommand{\@MANconclude}[4][1]{\@endflag{#1}\step*{#2}{#3}{#4}}

\newcommand{\@AUTOconclude}[4][1]{\@endflag{#1}\step{#2}{#3}{#4}}
```

5.2.4 The environment itself

flagderiv

This section explains the main environment for flag style derivations/proofs: flagderiv. The command has an optional argument [⟨*prefix*⟩], used as label prefix so that labels can be "scoped" with a namespace local to the derivation but still accessible from the outside.

Examples:
\begin{flagderiv} ... \end{flagderiv}
\begin{flagderiv}[abs] ... \end{flagderiv}

```
\newenvironment{flagderiv}[1][]{%
```

First, set the optional label prefix (if given) and reset the counters.

```
 \ifthenelse{\equal{#1}{}}{%
  \renewcommand{\@labelprefix}{}%
 }{%
  \renewcommand{\@labelprefix}{#1:}%
 }%
 \setcounter{fd@flagcount}{0}%
 \setcounter{fd@stepcount}{0}%
```

Note: *All* of the following five commands have a star version which allows manual numbering, instead of having auto-numbering, LaTeX labels and references.

\assume
\assume*

Opens an assumption flag; command usage:

\assume{⟨*label*⟩}{⟨*formula*⟩}{⟨*comment*⟩}
\assume*{⟨*custom-number*⟩}{⟨*formula*⟩}{⟨*comment*⟩}

Examples:
```
\assume{assum:1}{x > 0}{Assumption}
\assume*{l}{\neg P}{}
```

```
92 \newcommand{\assume}{\@ifstar{\@MANassume}{\@AUTOassume}}
```

`\introduce` `\introduce*` Opens an introduction flag. Command usage:

`\introduce{`⟨*label*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}`
`\introduce*{`⟨*custom-number*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}`

Examples:
```
\introduce{intro-x}{x \in \mathbb{N}}{Introduction}
\introduce*{l - 1}{P \in s_P}{}
```

```
93 \newcommand{\introduce}{\@ifstar{\@MANintroduction}{\@AUTOintroduction}}
```

`\conclude` `\conclude*` Performs a conclusion/multiple conclusion from one or more introductions and/or assumptions. The command usage:

`\conclude[`⟨*number*⟩`]{`⟨*label*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}`
`\conclude*[`⟨*number*⟩`]{`⟨*custon-number*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}`

The optional argument `[`⟨*number*⟩`]` indicates how much flags should be closed with this conclusion, this defaults to 1.

Examples;
```
\conclude[2]{concl:2}{x > 0 \implies ...}{$\implies$-intro on ...}
\conclude*{n - 1}{\neg\neg P}{$\implies$-intro on (l) ...}
```

```
94 \newcommand{\conclude}{\@ifstar{\@MANconclude}{\@AUTOconclude}}
```

`\step` `\step*` Performs a derivation step, command usage:

`\step{`⟨*label*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}`
`\step*{`⟨*custom-number*⟩`}{`⟨*formula*⟩`}{`⟨*comment*⟩`}`

Examples:
```
\step{x-elim}{x > 0}{$\forall$-elem on \ref{assum:1}}
\step*{n - 1}{x > 0}{$\forall$-elem on (n - 3)}
```

Note: `\\` (line breaks) may be inserted for splitting large formulas.

```
95 \newcommand{\step}{\@ifstar{\@MANstep}{\@AUTOstep}}
```

`\skipsteps` `\skipsteps*` Skips a number of steps (for usage in examples), usage:

`\skipsteps{`⟨*number*⟩`}{`⟨*formula*⟩`}{`⟨*comments*⟩`}`
`\skipsteps*{`⟨*formula*⟩`}{`⟨*comments*⟩`}`.

Examples:
```
\skipsteps{3}{\vdots}{Etcetera, etcetera}
\skipsteps*{\vdots}{Etcetera, etcetera}
```

```
96 \newcommand{\skipsteps}{\@ifstar{\@MANskipsteps}{\@AUTOskipsteps}}
```

`\done` Closes a number of flags because contexts have been opened and should be closed without conclusions. Command usage:

`\done[`⟨*number*⟩`]`

This command takes a number of introductions/assumptions, `[`⟨*number*⟩`]`, to close as an optional argument, which defaults to 1.

Examples:
`\done`
`\done[3]`

```
\newcommand{\done}{\@flagclose}
```

Silently override `\ref` to work with the defined namespace.

```
\let\origref\ref%
\renewcommand{\ref}[1]{\origref{\@labelprefix##1}}%
```

The actual flagderiv environment is just a longtable (called "derivation").

```
\renewcommand{\tablename}{Derivation}%
\begin{longtable}[l]{rll}
}{%
\end{longtable}
}
```

`flagderiv*` The `flagderiv*` environment is almost the same as `flagderiv` with the only difference that the resulting derivation will and can not be split across pages.

Example:
`\begin{flagderiv*} ... \end{flagderiv*}`

```
\newenvironment{flagderiv*}[1][]{%
 \ifthenelse{\equal{#1}{}}{%
  \renewcommand{\@labelprefix}{}%
 }{%
  \renewcommand{\@labelprefix}{#1:}%
 }%
 \setcounter{fd@flagcount}{0}%
 \setcounter{fd@stepcount}{0}%
 \newcommand{\assume}{\@ifstar{\@MANassume}{\@AUTOassume}}
 \newcommand{\introduce}{\@ifstar{\@MANintroduction}{\@AUTOintroduction}}
 \newcommand{\conclude}{\@ifstar{\@MANconclude}{\@AUTOconclude}}
 \newcommand{\step}{\@ifstar{\@MANstep}{\@AUTOstep}}
 \newcommand{\skipsteps}{\@ifstar{\@MANskipsteps}{\@AUTOskipsteps}}
 \newcommand{\done}{\@flagclose}
 \let\origref\ref%
 \renewcommand{\ref}[1]{\origref{\@labelprefix##1}}%
 \renewcommand{\tablename}{Derivation}
 \begin{tabular}{rll}
}{%
 \end{tabular}
}
```