# PTcalc Manual 

Kokichi Futatsugi

December 20, 2021

## 1 Overview

A CafeOBJ's specification (i.e. a module) $M$ contains equations and defines the set of constructor models $\mathbb{M o d}(M)$ each element of that satisfies the equations. PTcalc (Proof Tree Calculus) is a refined version of a CafeOBJ version of CITP (Constructor-based Inductive Theorem Prover) and helps to prove a property holds for any model in $\operatorname{Mod}(M)$. A property is described as a Boolean ground term $p$ with fresh constants that correspond to the parameters of the property.

If the term $p$ is deduced to be equal to true with the $M$ 's equations (in symbols $M \vdash_{\mathrm{e}} p$ ), any model of $M$ satisfies $p$ (in symbols $M \vDash p$ ) by the soundness of equational deduction (in symbols $M \vdash_{\mathrm{e}} p \Rightarrow M \models p$ ). If the term $p$ is reduced to true by using the $M$ 's equations as reduction (rewriting) rules from left to right (in symbols $M \vdash_{\mathrm{r}} p$ ), $M \vdash_{\mathrm{e}} p$ holds by the soundness of reduction with respect to equational deduction (in symbols $M \vdash_{\mathrm{r}} p \Rightarrow M \vdash_{\mathrm{e}} p$ ). Let $M \vdash_{\mathrm{c}} p$ denote that the CafeOBJ reduction command "red in $M$ : $p$." returns true. CafeOBJ's reduction is an implementation of $M \vdash_{r} p$ and we get $M \vdash_{\mathrm{c}} p \Rightarrow M \vdash_{\mathrm{r}} p$. Because implication $(\Rightarrow)$ is transitive the following proof rule is obtained.
(1) $\quad M \vdash_{\mathrm{c}} p \Rightarrow M \vDash p$

Usually $M \vdash_{\mathrm{c}} p$ is difficult to prove directly, and we need to find case splitting equations $e_{1}, \cdots, e_{n}$ such that at least one of them holds for any model in $\mathbb{M o d}(M)$. That is, the equations cover all the possibilities (i.e. are exhaustive). Let $M_{+e_{i}}$ be the module gotten by adding $e_{i}$ to $M$, then each model in $\operatorname{Mod}(M)$ is a model of $M_{+e_{j}}$ for some $j \in\{1,2, \cdots, n\}$, and we get the following proof rule of case split with exhaustive equations.

$$
\begin{equation*}
\left(M_{+e_{1}} \vDash p \wedge M_{+e_{2}} \vDash p \wedge \cdots \wedge M_{+e_{n}} \vDash p\right) \Rightarrow M \vDash p \tag{2}
\end{equation*}
$$

$M_{+e_{i}} \vdash_{\mathrm{c}} p$ would be still difficult to prove and (2) is applied repeatedly. The repeated applications of (2) generate proof trees successively. Each of the generated proof trees has the root node $M \models p$ and each of other nodes is of the form $M_{+e_{i_{1}} \cdots+e_{i_{m}}} \vDash p(m \in\{1,2, \cdots\})$ that is generated as a child node of $M_{+e_{i_{1}} \cdots+e_{i_{m-1}}} \vDash p$ by applying (2). A leaf node (i.e. a node without
child nodes) $M_{+e_{l_{1}} \cdots+e_{l_{k}}} \vDash p(k \in\{0,1, \cdots\})$ of a proof tree is called effective if $M_{+e_{l_{1}} \cdots+e_{l_{k}}} \vdash_{c} p$ holds. A proof tree is called effective if all of whose leaf nodes are effective. PTcalc proves $M \vDash p$ by constructing an effective proof tree whose root node is $M \models p$.

## 2 PTcalc Commands

Each of the PTcalc commands starts with the keyword : goal, : def, :apply, :csp, :ctf, :init, :red, :show, :describe, :select, or :set. The head character : distinguishes them from ordinary CafeOBJ commands.

In PTcalc, a node in a proof tree is a special CafeOBJ module and called a goal. Each goal $g l$ has a name like root or 1-2-3 (i.e. 3rd child of 2 nd child of 1 st child of root), and consists of the following five items.
(1) The next target (or default) goal Boolean tag NTG $(g l)$ that indicates a goal where a PTcalc command is executed. $(\operatorname{NTG}(g l)=$ true) holds for at most one goal $g l$ in a proof tree.
(2) The context module CTM ( $g l$ ) that is a CafeOBJ module and corresponts to $M$ of $M \vDash p$. The goal $g l$ inherits (imports) all the contents of CTM $(g l)$.
(3) The set of introduced axioms (i.e. assumptions) INA $(g l)$ that corresponds to $+e_{i_{1}} \cdots+e_{i_{m}}$ of $M_{+e_{i_{1}} \cdots+e_{i_{m}}} \vDash p$.
(4) The set of sentences (or equations) to be proved $\operatorname{STP}(g l)$ that corresponts to $p$ of $M_{+e_{i_{1}} \cdots+e_{i_{m}}} \vDash p$.
(5) The discharged Boolean tag $\operatorname{DCD}(g l)$ that indicates whether $g l$ is already discharged (i.e. proved).
$(\operatorname{CTM}(g l) \cup \operatorname{INA}(g l)) \vDash \operatorname{STP}(g l)$ corresponds to $M_{+e_{i_{1}} \cdots+e_{i_{m}}} \vDash p$, where $(\operatorname{CTM}(g l) \cup$ $\operatorname{INA}(g l))$ is understood as the module gotten by adding all the equations in $\operatorname{INA}(g l)$ to $\operatorname{CTM}(g l)$, and $\operatorname{STP}(g l)$ is understood as the conjunction of its elements. $g l$ sometimes means $(\operatorname{CTM}(g l) \cup I N A(g l))$.

The effects of PTcalc command executions are defined in the following subsections where a succession of commands ": cmnd ${ }_{1}:$ cmnd $_{2} "$ means executing $: c m n d_{2}$ after executing $: c m n d_{1}$.

## 2.1 :goal

For $n \in\{1,2, \cdots\}$ unconditional equations $e q_{i}(i \in\{1,2, \cdots, n\})$ a :goal command

$$
: \operatorname{goal}\left\{e q_{l} \quad e q_{2} \cdots e e q_{n}\right\}
$$

initiates a proof by generating a proof tree that consists only of the goal root as follows.
(1) $\operatorname{NTG}($ root $)=$ true.
(2) $\operatorname{CTM}$ (root) is the current module of CafeOBJ.
(3) INA (root) $=\{ \}$ (i.e. the empty set).
(4) $\operatorname{STP}($ root $)=\left\{e q_{l}, e q_{2}, \cdots, e q_{n}\right\}$.
(5) $\operatorname{DCD}($ root $)=$ false.

A CafeOBJ module $M$ becomes the current module by executing a CafeOBJ command "select $M$.". If some contents (i.e. sorts, operators, equations) are declared after a CafeOBJ command "open $M$. ", the opened tentative module $\%$ obtained by adding the contents to $M$ becomes the current module.

## 2.2 : def

A : def command gives a name to a :csp command, a :init command, or a sequence ( $\cdots$ ) of command names as follows.

```
:def csid = :csp...
:def inid = :init...
:def sqid = (id i id 2 \cdotsid
```

Each $i d_{i}(i \in\{1,2, \cdots, n\})$ is the built-in command name rd- or rd, or the already defined :csp command name csid, :init command name inid, or (...) command name sqid.

## 2.3 : apply $(c s i d)$ and $: \operatorname{csp}\{\cdots\}$ and $: \operatorname{ctf}\{$ eq $l=r$.

Let (a) $t g$ be a goal such that (NTG $(t g)=$ true) and (b) csid be the name of a : csp command defined as follows with $n \in\{1,2, \cdots\}$ equations $e q_{i}(i \in\{1,2, \cdots$, $n\}$ ).

```
:def csid = : csp{eq
```

Then executing the command
:apply (csid)
generates $n$ sub-goals (i.e. child goals) $t g-1, t g-2, \cdots, t g-\mathrm{n}$ of $t g$ as follows.
(1:1) Change NTG( $t g)$ from true to false.
(1:2) $\operatorname{NTG}(t g-1)=$ true.
(1:3) $\operatorname{NTG}(t g-i)=$ false $(i \in\{2, \cdots, n\})$.
(2) $\operatorname{CTM}(t g-i)=\operatorname{CTM}(t g) \quad(i \in\{1,2, \cdots, n\})$.
(3) $\operatorname{INA}(t g-i)=\operatorname{INA}(t g) \cup\left\{e q_{i}\right\} \quad(i \in\{1,2, \cdots, n\})$.
(4) $\operatorname{STP}(t g-i)=\operatorname{STP}(t g) \quad(i \in\{1,2, \cdots, n\})$.
(5) $\operatorname{DCD}(t g-i)=$ false $(i \in\{1,2, \cdots, n\})$.

Executing the :csp command

$$
: \operatorname{csp}\left\{e q_{1} e q_{2} \cdots e e q_{n}\right\}
$$

without giving a name has the same effect as defined above.
$: \operatorname{ctf}\{$ eq $l=r$.$\} is the abbreviation of$
$: \operatorname{csp}\{\mathrm{eq} l=r . \quad$ eq $(l=r)=$ false.$\}$.

## 2.4 : apply (inid) and ":init as eqid [lb] by $\{$ sbst $\}$ "

Let (a) $t g$ be a goal such that $(\operatorname{NTG}(t g)=$ true $),(\mathrm{b}) l b$ be a label of an equation in $t g$ (i.e. in CTM $(t g) \cup I N A(t g))$ with a set of labels lbset that includes $l b$, (c) inid be the name of a :init command defined as follows with a new equation name eqid and a substitution sbst, where sbst is a sequence of $n \in\{1,2, \cdots\}$ variable-term pairs " $v_{i}<-t_{i} ; "(i \in\{1,2, \cdots, n\})$.
: def inid $=$ :init as eqid [lb] by $\{s b s t\}$
"as eqid" can be omitted, [ $l b$ ] can be replaced with ( $e q$ ) by writing eq directly, and "by \{ sbst\}" can be omitted by replacing it with ".". Let "ceq $l=$ $r$ if $c$." be the equation with the label $l b$ or $e q$ (ceq stands for conditional equation). An unconditional equation is understood to have the condition true, i.e., $c=$ true.

Then executing the command
: apply (inid)
generates a sub-goal $t g$-1 of the goal $t g$ as follows, where $\hat{l}, \widehat{r}, \widehat{c}$ are normal forms of $s b s t(l), s b s t(r), s b s t(c)$ in the goal $t g$. If "by $\{s b s t\}$ " is omitted, $s b s t$ is understood to be the identity function. The equation "ceq $l=r$ if $c$." itself is not used to get the normal forms if it is declared with (...) (i.e. it is eq). The equation is, however, used if it is the equation with the label $l b$ and without :nonexec attribute, hence the :nonexec attribute should be declared if the label $l b$ is supposed to be used in a :init command. Note that, if the condition $c$ is true, using "ceq $l=r$ if $c$. ." for getting $\widehat{l}, \widehat{r}, \widehat{c}$ generates a trivial equation "ceq $\widehat{r}=\widehat{r}$ if true ." i.e., "eq $\widehat{r}=\widehat{r}$." (see $3: 1$ below).
(1:1) Change $\operatorname{NTG}(t g)$ from true to false.
(1:2) $\operatorname{NTG}\left(t^{\prime}-1\right)=$ true.
(2) $\operatorname{CTM}(t g-1)=\operatorname{CTM}(t g)$.
(3:0) $\operatorname{INA}(t g-1)=\operatorname{INA}(t g)$.
(4) $\operatorname{STP}(t g-1)=\operatorname{STP}(t g)$.
(5) $\operatorname{DCD}(t g-1)=$ false.
(3:1) Add "ceq[eqid : nonexec] : $\widehat{l}=\widehat{r}$ if $\widehat{c}$." to $\operatorname{INA}(\operatorname{tg}-1)$
if $(\widehat{c}=\mathrm{false}) \vee(\widehat{l}=\widehat{r})$.
(3:2) Add "eq[eqid] : $\widehat{r}=\widehat{l} . "$ to $\operatorname{INA}(\operatorname{tg}-1)$
if $(\widehat{c}=$ true $) \wedge \neg(\widehat{l}=\widehat{r}) \wedge((\widehat{l}=$ true $) \vee(\widehat{l}=\mathrm{false}))$.
(3:3) Add "eq[eqid]: $\widehat{l}=\widehat{r} . "$ to $\operatorname{INA}(t g-1)$
if $(\widehat{c}=$ true $) \wedge \neg(\widehat{l}=\widehat{r}) \wedge \neg((\widehat{l}=$ true $) \vee(\widehat{l}=\mathrm{false}))$.
(3:4) Add "ceq[eqid]: $\widehat{r}=\widehat{l}$ if $\widehat{c} . "$ to $\operatorname{INA}(t g-1)$
if $\neg((\widehat{c}=$ true $) \vee(\widehat{c}=$ false $)) \wedge \neg(\widehat{l}=\widehat{r}) \wedge$
$((\widehat{l}=$ true $) \vee(\widehat{l}=$ false $))$.
(3:5) Add "ceq[eqid]: $\widehat{l}=\widehat{r}$ if $\widehat{c} . "$ to $\operatorname{INA}(t g-1)$
if $\neg((\widehat{c}=$ true $) \vee(\widehat{c}=$ false $)) \wedge \neg(\widehat{l}=\widehat{r}) \wedge$
$\neg((\widehat{l}=\mathrm{true}) \vee(\widehat{l}=\mathrm{false}))$.
Executing the :init command :init as eqid [ $l b$ ] by $\{s b s t\}$
without giving a name has the same effect as defined above.

## 2.5 : apply (rd-)

Let (a) $t g$ be the goal such that $(\operatorname{NTG}(t g)=$ true $)$ and (b) $\operatorname{STP}(t g)=\left\{e q_{1}\right.$, $\left.e q_{2}, \cdots, e q_{n}\right\}(n \in\{1,2, \cdots\})$ with $l_{i}$ and $r_{i}$ being the left hand and the right hand of $e q_{i}(i \in\{1, \cdots, n\})$. Let a goal $g l-s$ be a sibling goal of a goal $g l$ if $g l-s$ and $g l$ are child goals of the same goal.

Then executing the command
: apply (rd-)
changes $\operatorname{STP}(t g), \operatorname{DCD}\left(\_\right), \operatorname{NTG}\left(\_\right)$with the following procedure. If INA $(t g)$ contains executable (i.e. not having :nonexec attribute) "eq true = false ." or "eq false = true .", the contradictory equation has been generated with the equations in $\operatorname{CTM}(t g) \cup I N A(t g)$ in the execution of :init command (see 2.4) and the goal $t g$ can be discharged.

```
IF (INA \((t g)\) does not contain executable
    "eq true = false ." or "eq false = true .")
    THEN
    [ For \(i \in\{1, \cdots, n\}\) let \(\widehat{e q_{i}}\) be the normal form of \(\left(l_{i}=r_{i}\right)\) in \(t g\) ];
    [(4) For \(i \in\{1, \cdots, n\}\) erase \(e q_{i}\) from \(\operatorname{STP}(t g)\) if ( \(\widehat{e q_{i}}=\) true \()\) ];
    IF \(\operatorname{not}(\operatorname{STP}(t g)=\{ \})\) THEN STOP FI
    FI ;
    [ (5:1) Change \(\operatorname{DCD}(t g)\) to true ] ;
    (1:1) Change NTG \((t g)\) to false ] ;
    [ Let \(t m p\) be \(t g\) ] ;
    WHILE \(((\operatorname{DCD}(t m p-s)=\) true \()\) for any sibling goal \(t m p-s\) of \(t m p)\)
    DO
    [ (5:2) Change \(\operatorname{DCD}(t m p-p)\) to true for the parent goal tmp-p of tmp] ;
    [ Let \(t m p\) be \(t m p-p\) ]
    OD ;
    [ \((1: 2)\) Change \(\operatorname{NTG}(l f)\) to true for the goal \(l f\) that is the first leaf goal
    in the lexicographic order such that \((\operatorname{DCD}(l f)=f a l s e)\)
    if such goal lf exists ]
```


## 2.6 : apply (rd)

Let (a) $t g$ be the goal such that $\left(\operatorname{NTG}(t g)=\right.$ true) and (b) $\operatorname{STP}(t g)=\left\{e q_{1}\right.$, $\left.e q_{2}, \cdots, e q_{n}\right\} \quad(n \in\{1,2, \cdots\})$ with $l_{i}$ and $r_{i}$ being the left hand and the right hand of $e q_{i}(i \in\{1, \cdots, n\})$.

Then executing the command
:apply(rd)
generates a sub-goal $t g-1$ of the goal $t g$ and changes $\operatorname{STP}(\operatorname{tg}-1), \operatorname{DCD}\left(\_\right), \operatorname{NTG}\left(\_\right)$ as follows.
(1:1) Change $\operatorname{NTG}(t g)$ from true to false.
(1:2) $\operatorname{NTG}\left(t^{\prime}-1\right)=$ true.
(2) $\operatorname{CTM}(\operatorname{tg}-1)=\operatorname{CTM}(t g)$.
(3) $\operatorname{INA}(t g-1)=\operatorname{INA}(t g)$.
(4:1) $\operatorname{STP}(t g-1)=\operatorname{STP}(t g)$.
(5:1) $\operatorname{DCD}(t g-1)=$ false.
IF (INA ( $t g-1$ ) does not contain executable
"eq true $=$ false.$"$ or "eq false $=$ true .")
THEN
[ For $i \in\{1, \cdots, n\}$ let $\widehat{l_{i}}, \widehat{r_{i}}, \widehat{e q_{i}}$ be the normal forms
of $l_{i}, r_{i},\left(l_{i}=r_{i}\right)$ in $\left.t g-1\right]$;
$\left[(4: 2)\right.$ For each $e q_{i} \in \operatorname{STP}(t g-1)(i \in\{1, \cdots, n\})$
if $\left(\widehat{e q_{i}}=\right.$ true $)$ then erase $e q_{i}$
else replace $e q_{i}$ with "eq $\left.\widehat{l_{i}}=\widehat{r_{i}} . " \mathrm{fi}\right]$;

IF $\operatorname{not}(\operatorname{STP}(t g-1)=\{ \})$ THEN STOP FI FI ;
$[(5: 2)$ Change $\operatorname{DCD}(\operatorname{tg}-1)$ to true ];
$[(1: 3)$ Change $\operatorname{NTG}(t g-1)$ to false ];
[ Let $t m p$ be $t g-1$ ] ;
WHILE $((\operatorname{DCD}(t m p-s)=$ true $)$ for any sibling goal tmp-s of tmp)
DO
[ (5:3) Change $\operatorname{DCD}(t m p-p)$ to true for the parent goal tmp-p of $t m p]$; [ Let $t m p$ be $t m p-p$ ]
OD ;
[ (1:4) Change NTG $(l f)$ to true for the goal $l f$ that is the first leaf goal in the lexicographic order such that $(\operatorname{DCD}(l f)=f a l s e)$
if such goal lf exists ]

## 2.7 : select

Let $g l$ and $t g$ be goals such that $(\operatorname{NTG}(t g)=$ true $)$. The command :select $g l$. changes $\operatorname{NTG}(t g)$ to false and $\operatorname{NTG}(g l)$ to true.

## 2.8 : apply $\left(i d_{1} i d_{2} \cdots i d_{n}\right)$

Let $t g$ be a goal such that $(\operatorname{NTG}(t g)=$ true $)$, and let each $i d_{i}(i \in\{1,2, \cdots n\})$ be the built-in command name rd- or rd, or the defined name of a :csp $\cdots$, :init..., or ( $\cdots$ ) command (see 2.2). Then the effect of the execution of : apply $\left(i d_{1} i d_{2} \cdots i d_{n}\right)$
on $\operatorname{NTG}\left({ }_{-}\right), \operatorname{CTM}(-), \operatorname{INA}(-), \operatorname{STP}(-), \operatorname{DCD}(-)$ is defined as follows.

### 2.8.1 $i d_{1}=:$ csp $\cdots$

Let $i d_{1}$ be the name of a :csp command defined as

```
:def id ( }=:\operatorname{csp}{e\mp@subsup{q}{1}{}\quade\mp@subsup{q}{2}{}\cdots\cdotse\mp@subsup{q}{m}{}}\quad(1\leqm
```

then

```
:apply(id }i\mp@subsup{|}{2}{}\cdotsi\mp@subsup{d}{n}{})
```

: apply ( $i d_{1}$ )
: select $t g-1 .: \operatorname{apply}\left(i d_{2} \cdots i d_{n}\right)$
:select $t g-2 .: \operatorname{apply}\left(i d_{2} \cdots i d_{n}\right)$
...
: select $t g-m .: \operatorname{apply}\left(i d_{2} \cdots i d_{n}\right)$.

### 2.8.2 $i d_{1}=$ :init...

Let $i d_{1}$ be the name of a :init command defined as :def $i d_{1}=$ :init...
then
:apply $\left(i d_{1} i d_{2} \cdots i d_{n}\right)=: \operatorname{apply}\left(i d_{1}\right): \operatorname{apply}\left(i d_{2} \cdots i d_{n}\right)$.

### 2.8.3 $\quad i d_{1}=(\cdots)$

Let $i d_{1}$ be the name of a ( $\cdots$ ) command defined as
:def $i d_{1}=\left(i d s_{1} i d s_{2} \cdots i d s_{m}\right) \quad(m \in\{1,2, \cdots\})$
then
: apply $\left(i d_{1} i d_{2} \cdots i d_{n}\right)=: \operatorname{apply}\left(i d s_{1} i d s_{2} \cdots i d s_{m} i d_{2} \cdots i d_{n}\right)$.
2.8.4 $i d_{1}=r d-$ or $r d$

Let (a) $t g$ be the goal such that $(\operatorname{NTG}(t g)=$ true $)$ and (b) $i d_{1}$ be rd- or rd.
If $\operatorname{DCD}(t g)$ is true after executing : apply $\left(i d_{1}\right)$ then
: apply $\left(i d_{1} i d_{2} \cdots i d_{n}\right)=: \operatorname{apply}\left(i d_{1}\right)$.
If $\operatorname{DCD}(t g)$ is not true after executing : apply $\left(i d_{1}\right)$ then
:apply $\left(i d_{1} i d_{2} \cdots i d_{n}\right)=: \operatorname{apply}\left(i d_{1}\right): \operatorname{apply}\left(i d_{2} \cdots i d_{n}\right)$.

## 2.9 :show and :describe

The :show and :describe commands show the status of the current proof tree. :show can be abbreviated as :sh, and :describe can be abbreviated as :desc.
$\triangleright$ :show proof
shows all the names of goals in the current proof tree with NTG (_) and DCD (_).
$\triangleright$ :show unproved shows $\operatorname{CTM}\left({ }_{-}\right)$, $\left.\operatorname{INA}()_{-}\right), \operatorname{STP}\left(\_\right), \operatorname{DCD}\left({ }_{-}\right)$for all the leaf goals in the current proof tree such that $(\operatorname{DCD}(-)=\mathrm{false})$.
$\triangleright$ :show discharged shows the discharged sentence with its context module.
$\triangleright$ :show goal
shows $\operatorname{CTM}(g l), \operatorname{INA}(g l), \operatorname{STP}(g l), \operatorname{DCD}(g l)$ for the goal $g l$ such that $(\operatorname{NTG}(g l)$ $=$ true).
$\triangleright$ :show goal gl shows $\operatorname{CTM}(g l), \operatorname{INA}(g l), \operatorname{STP}(g l), \operatorname{DCD}(g l)$ for the goal $g l$.
$\triangleright$ :show def
shows all the command names defined with : def commands.
$\triangleright$ :describe proof
shows $\operatorname{CTM}\left(\_\right)$, $\operatorname{INA}\left(\_\right), \operatorname{STP}\left(\_\right), \operatorname{DCD}\left(\_\right)$for all the goals in the current proof tree.

### 2.10 :red

$\triangleright$ :red trm.
shows the normal form of the term $\operatorname{trm}$ in the goal $t g$ such that (NTG $(t g)$ = true).
$\triangleright$ :red in $g l$ : trm. shows the normal form of the term $t r m$ in the goal gl .

### 2.11 : set

$\triangleright$ :set(verbose,on)
sets the PTcalc flag verbose on, and makes outputs from PTcalc more detailed.
$\triangleright$ :set(verbose,off) sets the PTcalc flag verbose off.
$\triangleright$ :set(verbose,show) shows the value of the PTcalc flag verbose.

