On sequent calculi vs natural deductions in logic and computer science

L. Gordeev

Uni-Tübingen, Uni-Ghent, PUC-Rio

PUC-Rio, Rio de Janeiro, October 13, 2015

§1. Sequent calculus (SC): Basics -1-

- Gentzen invented *sequent calculus* in order to prove Hilbert's consistency (more precisely, contradiction-free) assertion for pure logic and Peano Arithmetic. He succeeded in both cases, although the latter proof required consistency of Cantor's basic system of ordinals below ε_0 .

§1. Sequent calculus (SC): Basics -1-

- Gentzen invented *sequent calculus* in order to prove Hilbert's consistency (more precisely, contradiction-free) assertion for pure logic and Peano Arithmetic. He succeeded in both cases, although the latter proof required consistency of Cantor's basic system of ordinals below ε_0 .
- To this end he replaced a familiar Hilbert-style logic formalism based on the rule of detachment (aka *modus ponens*)

$$\boxed{\dfrac{\alpha \qquad \alpha \rightarrow \beta}{\beta}}$$

by a system R of *direct inferences* having *subformula property*: 'premise formulas occur as (sub)formulas in the conclusion'.

- Gentzen invented *sequent calculus* in order to prove Hilbert's consistency (more precisely, contradiction-free) assertion for pure logic and Peano Arithmetic. He succeeded in both cases, although the latter proof required consistency of Cantor's basic system of ordinals below ε_0 .
- To this end he replaced a familiar Hilbert-style logic formalism based on the rule of detachment (aka *modus ponens*)

$$\boxed{\dfrac{\alpha \qquad \alpha \to \beta}{\beta}}$$

by a system R of *direct inferences* having *subformula property*: 'premise formulas occur as (sub)formulas in the conclusion'.
- Such R (finitary, generally well-founded) is consistent, since $\perp$ (or $0 = 1$) has no proper subformula, and hence not derivable.

- To complete the consistency proof it remains to show that modus ponens is admissible in S.

§1. Sequent calculus: Basics -2-

- To complete the consistency proof it remains to show that modus ponens is admissible in S.
- In sequent form, modus ponens is called *cut* and looks like this

$$\boxed{\frac{\Gamma \Rightarrow \alpha \qquad \Gamma, \alpha \Rightarrow \beta}{\Gamma \Rightarrow \beta}} \ (\mathit{int.}) \ \ \mathit{or} \ \boxed{\frac{\Gamma, \alpha \qquad \Gamma, \neg\alpha}{\Gamma}} \ (\mathit{class.})$$

§1. Sequent calculus: Basics -2-

- To complete the consistency proof it remains to show that modus ponens is admissible in S.
- In sequent form, modus ponens is called *cut* and looks like this

$$\boxed{\dfrac{\Gamma \Rightarrow \alpha \qquad \Gamma, \alpha \Rightarrow \beta}{\Gamma \Rightarrow \beta}} \ (int.) \ \ or \ \boxed{\dfrac{\Gamma, \alpha \qquad \Gamma, \neg\alpha}{\Gamma}} \ (class.)$$

- So *cut elimination theorem* does the job.

- To complete the consistency proof it remains to show that modus ponens is admissible in S.
- In sequent form, modus ponens is called *cut* and looks like this

$$\boxed{\dfrac{\Gamma \Rightarrow \alpha \qquad \Gamma, \alpha \Rightarrow \beta}{\Gamma \Rightarrow \beta}} \; (int.) \;\; or \;\; \boxed{\dfrac{\Gamma, \alpha \qquad \Gamma, \neg\alpha}{\Gamma}} \; (class.)$$

- So *cut elimination theorem* does the job.

Theorem (cut elimination)

- To complete the consistency proof it remains to show that modus ponens is admissible in S.
- In sequent form, modus ponens is called *cut* and looks like this

$$\boxed{\frac{\Gamma \Rightarrow \alpha \qquad \Gamma, \alpha \Rightarrow \beta}{\Gamma \Rightarrow \beta}}\ (int.)\ \ or\ \boxed{\frac{\Gamma, \alpha \qquad \Gamma, \neg\alpha}{\Gamma}}\ (class.)$$

- So *cut elimination theorem* does the job.

Theorem (cut elimination)

1. *Logic: Every sequent derivable in* $R \cup \{cut\}$ *is derivable in* R .

- To complete the consistency proof it remains to show that modus ponens is admissible in S.
- In sequent form, modus ponens is called *cut* and looks like this

$$\boxed{\dfrac{\Gamma \Rightarrow \alpha \qquad \Gamma, \alpha \Rightarrow \beta}{\Gamma \Rightarrow \beta}} \ (int.) \ \ or \ \boxed{\dfrac{\Gamma, \alpha \qquad \Gamma, \neg\alpha}{\Gamma}} \ (class.)$$

- So *cut elimination theorem* does the job.

Theorem (cut elimination)

1. *Logic: Every sequent derivable in $R \cup \{cut\}$ is derivable in R .*
2. *Peano Arithmetic: Every qf-sequent derivable in $R_{PA} \cup \{cut\}$ is derivable in R_{PA} .*

- Due to Kreisel's observation one can use cut elimination techniques to establish *proof-theoretic conservations* : ' every formula provable in T is provable in sub-theory S '.

§1.1. Sequent calculus: Conservative extensions

- Due to Kreisel's observation one can use cut elimination techniques to establish *proof-theoretic conservations* : ' every formula provable in T is provable in sub-theory S '.
- The trick: express syntax a/o axioms of $T \setminus S$ using appropriate cuts which can be eliminated from sequent calculus of T.

§1.1. Sequent calculus: Conservative extensions

- Due to Kreisel's observation one can use cut elimination techniques to establish *proof-theoretic conservations* : ' every formula provable in T is provable in sub-theory S '.
- The trick: express syntax a/o axioms of $T \setminus S$ using appropriate cuts which can be eliminated from sequent calculus of T.

Example (ACA_0 *is conservative extension of* PA)

- Due to Kreisel's observation one can use cut elimination techniques to establish *proof-theoretic conservations* : ' every formula provable in T is provable in sub-theory S '.
- The trick: express syntax a/o axioms of $T \setminus S$ using appropriate cuts which can be eliminated from sequent calculus of T.

Example (ACA$_0$ *is conservative extension of* PA)

Every 1-order formula provable in ACA_0 is provable in PA, where ACA_0 extends PA by adding 2-order set-variables together with (corresponding logic and) axioms for 1-order comprehension and induction restricted to sets.

- Schütte (and followers) generalized Gentzen's arithmetical consistency proof working with infinite well-founded tree-like derivations supplied with ordinal labels. This yields deeper insight into proof-theoretic ordinals.

- Schütte (and followers) generalized Gentzen's arithmetical consistency proof working with infinite well-founded tree-like derivations supplied with ordinal labels. This yields deeper insight into proof-theoretic ordinals.
- Namely, for much stronger than PA theories T it's possible to describe proof-theoretic ordinals $\alpha_T >> \varepsilon_0$ which characterize theorems of T as follows:

§1.2. Sequent calculus: Ordinal analysis and beyond

- Schütte (and followers) generalized Gentzen's arithmetical consistency proof working with infinite well-founded tree-like derivations supplied with ordinal labels. This yields deeper insight into proof-theoretic ordinals.
- Namely, for much stronger than *PA* theories T it's possible to describe proof-theoretic ordinals $\alpha_T >> \varepsilon_0$ which characterize theorems of T as follows: ' every arithmetical theorem of T is provable in *PA* extended by transfinite induction below α_T '.

- Schütte (and followers) generalized Gentzen's arithmetical consistency proof working with infinite well-founded tree-like derivations supplied with ordinal labels. This yields deeper insight into proof-theoretic ordinals.
- Namely, for much stronger than *PA* theories T it's possible to describe proof-theoretic ordinals $\alpha_T >> \varepsilon_0$ which characterize theorems of T as follows: ' every arithmetical theorem of T is provable in *PA* extended by transfinite induction below α_T '.
- More recent research (initiated by Harvey Friedman) enables us to replace ordinals α_T (which are very involved for strong T) by more transparent quasi-ordinals characterized by extended Kruskal-style tree theorems.

§1.2. Sequent calculus: Ordinal analysis and beyond

- Schütte (and followers) generalized Gentzen's arithmetical consistency proof working with infinite well-founded tree-like derivations supplied with ordinal labels. This yields deeper insight into proof-theoretic ordinals.
- Namely, for much stronger than *PA* theories T it's possible to describe proof-theoretic ordinals $\alpha_T >> \varepsilon_0$ which characterize theorems of T as follows: ' every arithmetical theorem of T is provable in *PA* extended by transfinite induction below α_T '.
- More recent research (initiated by Harvey Friedman) enables us to replace ordinals α_T (which are very involved for strong T) by more transparent quasi-ordinals characterized by extended Kruskal-style tree theorems.
- This stuff is obviously related to (say, extended) Hilbert's Program concerning logic foundations of mathematics.

To put it in a nutshell, cut elimination provides an extremely powerful tool in Hilbert-style proof theory. Moreover it is constructive, and hence yields by now strongest conservation results for various intuitionistic theories [L.G]. Besides, it enables to work with cutfree systems of direct sequent rules. Also note:

To put it in a nutshell, cut elimination provides an extremely powerful tool in Hilbert-style proof theory. Moreover it is constructive, and hence yields by now strongest conservation results for various intuitionistic theories [L.G]. Besides, it enables to work with cutfree systems of direct sequent rules. Also note:

- Cutfree sequent calculi have better proof search.

§1.3. Sequent calculus: Some consequences

To put it in a nutshell, cut elimination provides an extremely powerful tool in Hilbert-style proof theory. Moreover it is constructive, and hence yields by now strongest conservation results for various intuitionistic theories [L.G]. Besides, it enables to work with cutfree systems of direct sequent rules. Also note:

- Cutfree sequent calculi have better proof search.
- However, there are complexity problems (re: speed-up).

To put it in a nutshell, cut elimination provides an extremely powerful tool in Hilbert-style proof theory. Moreover it is constructive, and hence yields by now strongest conservation results for various intuitionistic theories [L.G]. Besides, it enables to work with cutfree systems of direct sequent rules. Also note:

- Cutfree sequent calculi have better proof search.
- However, there are complexity problems (re: speed-up).
- What to do? Dag-like cutfree derivations and substitution rule!

§1.3. Sequent calculus: Some consequences

To put it in a nutshell, cut elimination provides an extremely powerful tool in Hilbert-style proof theory. Moreover it is constructive, and hence yields by now strongest conservation results for various intuitionistic theories [L.G]. Besides, it enables to work with cutfree systems of direct sequent rules. Also note:

- Cutfree sequent calculi have better proof search.
- However, there are complexity problems (re: speed-up).
- What to do? Dag-like cutfree derivations and substitution rule!
- Full dag-like compression with substitution may provide a solution (at least in the propositional case).

To put it in a nutshell, cut elimination provides an extremely powerful tool in Hilbert-style proof theory. Moreover it is constructive, and hence yields by now strongest conservation results for various intuitionistic theories [L.G]. Besides, it enables to work with cutfree systems of direct sequent rules. Also note:

- Cutfree sequent calculi have better proof search.
- However, there are complexity problems (re: speed-up).
- What to do? Dag-like cutfree derivations and substitution rule!
- Full dag-like compression with substitution may provide a solution (at least in the propositional case).
- But main complexity problem remains open (re: NP vs coNP).

Prawitz formalized (Gentzen's notion of) *natural deductions*.
Here's a short overview of ND features (compared to SC).

§2. Natural deductions (ND): Basic comparison -1-

Prawitz formalized (Gentzen's notion of) *natural deductions*.
Here's a short overview of ND features (compared to SC).

1. Lambda calculus connections. (Disadvantage: geometry fails.)

§2. Natural deductions (ND): Basic comparison -1-

Prawitz formalized (Gentzen's notion of) *natural deductions*.
Here's a short overview of ND features (compared to SC).

1. Lambda calculus connections. (Disadvantage: geometry fails.)
2. Normalization (also constructive) instead of cut elimination. Normal ND also have global subformula property (but, unlike cutfree SC, not the local one).

§2. Natural deductions (ND): Basic comparison -1-

Prawitz formalized (Gentzen's notion of) *natural deductions*.
Here's a short overview of ND features (compared to SC).

1. Lambda calculus connections. (Disadvantage: geometry fails.)
2. Normalization (also constructive) instead of cut elimination. Normal ND also have global subformula property (but, unlike cutfree SC, not the local one).
   Remember that normal ND are not like cutfree SC; actually they include modus ponens explicitly.

§2. Natural deductions (ND): Basic comparison -1-

Prawitz formalized (Gentzen's notion of) *natural deductions*.
Here's a short overview of ND features (compared to SC).

1. Lambda calculus connections. (Disadvantage: geometry fails.)
2. Normalization (also constructive) instead of cut elimination. Normal ND also have global subformula property (but, unlike cutfree SC, not the local one).
   Remember that normal ND are not like cutfree SC; actually they include modus ponens explicitly.
3. ND normalization is weaker than SC cut elimination. And/but it also features exponential speed-up.

Prawitz formalized (Gentzen's notion of) *natural deductions*.
Here's a short overview of ND features (compared to SC).

1. Lambda calculus connections. (Disadvantage: geometry fails.)
2. Normalization (also constructive) instead of cut elimination. Normal ND also have global subformula property (but, unlike cutfree SC, not the local one).
   Remember that normal ND are not like cutfree SC; actually they include modus ponens explicitly.
3. ND normalization is weaker than SC cut elimination. And/but it also features exponential speed-up.
4. Ordinal analysis?

§2. Natural deductions (ND): Basic comparison -1-

Prawitz formalized (Gentzen's notion of) *natural deductions*. Here's a short overview of ND features (compared to SC).

1. Lambda calculus connections. (Disadvantage: geometry fails.)
2. Normalization (also constructive) instead of cut elimination. Normal ND also have global subformula property (but, unlike cutfree SC, not the local one).
   Remember that normal ND are not like cutfree SC; actually they include modus ponens explicitly.
3. ND normalization is weaker than SC cut elimination. And/but it also features exponential speed-up.
4. Ordinal analysis?
5. Stronger ties to complexity theory?

§2. Natural deductions: Basic comparison -2-

- Due to lack of time let's go straight to final clause 5.

§2. Natural deductions: Basic comparison -2-

- Due to lack of time let's go straight to final clause 5.
- Main question: How to get a short propositional ND?

- Due to lack of time let's go straight to final clause 5.
- Main question: How to get a short propositional ND?
- Recall that sequent derivations (proofs) admit full dag-like compressions. (In the sequel we consider propositional logic.)

- Due to lack of time let's go straight to final clause 5.
- Main question: How to get a short propositional ND?
- Recall that sequent derivations (proofs) admit full dag-like compressions. (In the sequel we consider propositional logic.)

Theorem (L.G.)

§2. Natural deductions: Basic comparison -2-

- Due to lack of time let's go straight to final clause 5.
- Main question: How to get a short propositional ND?
- Recall that sequent derivations (proofs) admit full dag-like compressions. (In the sequel we consider propositional logic.)

Theorem (L.G.)

Any given tree-like sequent proof T (whether cutfree or not) is constructively compressible to a dag-like sequent proof D of the same endsequent such that the total number of pairwise distinct nodes in D is less, or equal, than the total number of pairwise distinct sequents occurring in T. Loosely speaking this holds also in the presence of substitution rule(s).

§2. Natural deductions: Basic comparison -3-

§2. Natural deductions: Basic comparison -3-

- Clearly this result yields a significant space reduction.

- Clearly this result yields a significant space reduction.
- But there is one catch: We can't properly control the number of pairwise distinct sequents (as being sets of formulas) occurring in T even if we know that all formulas in question are subformulas of the conclusion.

- Clearly this result yields a significant space reduction.
- But there is one catch: We can't properly control the number of pairwise distinct sequents (as being sets of formulas) occurring in T even if we know that all formulas in question are subformulas of the conclusion.
  It's still exponential upper bound! What to do?

§2. Natural deductions: Basic comparison -3-

- Clearly this result yields a significant space reduction.
- But there is one catch: We can't properly control the number of pairwise distinct sequents (as being sets of formulas) occurring in T even if we know that all formulas in question are subformulas of the conclusion.
  It's still exponential upper bound! What to do?
- Recall that, by contrast, ND's consist of single formulas. Moreover, the normal ones share the same subformula property. Is it possible to compress them analogously and obtain polynomial upper bounds on the total number of nodes?

- Warning:
  There is one obstacle when it comes to compression of natural deductions. Namely, we should avoid *vertical compressions* due to possible confusion caused by discharged assumptions. (This problem is irrelevant to proof compression theory in sequent calculi).

§2. Natural deductions: Basic comparison -4-

- Warning:
  There is one obstacle when it comes to compression of natural deductions. Namely, we should avoid *vertical compressions* due to possible confusion caused by discharged assumptions. (This problem is irrelevant to proof compression theory in sequent calculi).
- What can be done is a sort of *horizontal dag-like compression*.

- Warning:
  There is one obstacle when it comes to compression of natural deductions. Namely, we should avoid *vertical compressions* due to possible confusion caused by discharged assumptions. (This problem is irrelevant to proof compression theory in sequent calculi).
- What can be done is a sort of *horizontal dag-like compression*.
- Underlying idea:
  If an input tree-like deduction has merely polynomial height and every horizontal section is fully dag-like compressible (i.e. reducible to pairwise distinct formulas), then the resulting dag-like deduction has polynomial size. Voilà!

§2.1. Dag-like natural deductions -1-

- The assumptions on the tree-like complexity in question can be obtained by embedding into ND calculus Hudelmaier's sequent calculus for minimal a/o intuitionistic logic (without disjunction).

- The assumptions on the tree-like complexity in question can be obtained by embedding into ND calculus Hudelmaier's sequent calculus for minimal a/o intuitionistic logic (without disjunction).
- All in all such dag-like compression will infer **NP** = **PSPACE**.

- The assumptions on the tree-like complexity in question can be obtained by embedding into ND calculus Hudelmaier's sequent calculus for minimal a/o intuitionistic logic (without disjunction).
- All in all such dag-like compression will infer **NP = PSPACE**.
- But to this end we have to formalize dag-like deducibility in Prawitz's world. Recall that *'dag'* stands for *directed acyclic graph* (edges directed upwards).

- The assumptions on the tree-like complexity in question can be obtained by embedding into ND calculus Hudelmaier's sequent calculus for minimal a/o intuitionistic logic (without disjunction).
- All in all such dag-like compression will infer **NP** = **PSPACE**.
- But to this end we have to formalize dag-like deducibility in Prawitz's world. Recall that *'dag'* stands for *directed acyclic graph* (edges directed upwards).
- The main difference between tree-like and dag-like natural deductions is caused by the art of discharging, as the following example shows.

§2.1. Dag-like natural deductions -2-

Example

Example

Consider a dag-like natural deduction $\partial =$

$$(\to \mathrm{E})\ \dfrac{(\to \mathrm{E})\ \dfrac{\dfrac{\begin{matrix}\overline{\ \Gamma\ }\\ \overbrace{\qquad}\\ \vdots\end{matrix}}{\beta \to \alpha} \qquad \dfrac{[\alpha]^1 \qquad \alpha \to \beta}{\beta}\ (\to \mathrm{E})}{\alpha \qquad\qquad\qquad \alpha \to \beta^{[1]}}\ (\to \mathrm{I})}{\beta}$$

in which the right-hand side premise of second $(\to \mathrm{E})$ coincides with $(\to \mathrm{I})$ premise β. Note that the assumption α above β is discharged by this $(\to \mathrm{I})$. However, we can only infer that ∂ deduces β from $\Gamma \cup \{\alpha, \alpha \to \beta\}$, instead of expected $\Gamma \cup \{\alpha \to \beta\}$, which leaves the option $\Gamma \cup \{\alpha \to \beta\} \nvdash \beta$ open, if $\alpha \notin \Gamma$.

Example (*continued*)

Example (*continued*)

This becomes obvious if we replace ∂ by its "unfolded" tree-like version $\partial_{\mathrm{U}} =$

$$\dfrac{\dfrac{\dfrac{\alpha \qquad \alpha\to\beta}{\beta} \qquad \dfrac{\begin{array}{c}\Gamma\\ \hline \overbrace{\qquad\quad}\\ \vdots\end{array}}{\beta\to\alpha}}{\alpha} \qquad \dfrac{\dfrac{[\alpha]^1 \qquad \alpha\to\beta}{\beta}}{\alpha\to\beta}\,[1]}{\beta}$$

Clearly ∂_{U} deduces β from $\Gamma \cup \{\alpha, \alpha \to \beta\}$, instead of $\Gamma \cup \{\alpha \to \beta\}$, which leaves the option $\Gamma \cup \{\alpha \to \beta\} \nvdash \beta$ open, if $\alpha \notin \Gamma$.

[1] unless ∂ is tree-like, in which case both propeties are in $\mathbf{P}$.

- Keeping this in mind we'll say that in a dag-like natural deduction ∂, a given leaf *u* is an *open* (or *undischarged*) *assumption-node* iff there exists a thread θ connecting *u* with the root and such that no $w \in \theta$ is the $(\to \mathrm{I})$ conclusion assigned with $\alpha \to \beta$, provided that α is assigned to *u*. Other leaves are called *closed* (or *discharged*) in ∂. Note that the corresponding condition '*u is open (resp. closed) in* ∂' belongs merely to **NP** (resp. **coNP**) [1], and hence ad hoc is inappropriate for polynomial time proof verification.

[1] unless ∂ is tree-like, in which case both propeties are in **P**.

§2.1. Dag-like natural deductions -4-

- Keeping this in mind we'll say that in a dag-like natural deduction ∂, a given leaf u is an *open* (or *undischarged*) *assumption-node* iff there exists a thread θ connecting u with the root and such that no $w \in \theta$ is the $(\rightarrow \mathrm{I})$ conclusion assigned with $\alpha \rightarrow \beta$, provided that α is assigned to u. Other leaves are called *closed* (or *discharged*) in ∂. Note that the corresponding condition '*u is open (resp. closed) in ∂*' belongs merely to **NP** (resp. **coNP**) [1], and hence ad hoc is inappropriate for polynomial time proof verification.
- We overcome this trouble by a suitable modification of the notion of local correctness that includes special vertex-labeling function $\ell^{\mathrm{d}} : \mathrm{V}(D) \times \mathrm{F}(D) \rightarrow \{0, 1\}$, where $\mathrm{V}(D)$ and $\mathrm{F}(D)$ are respectively the vertices ($=$ nodes) and formulas of the underlying dag D.

[1] unless ∂ is tree-like, in which case both propeties are in **P**.

- Other two (standard) labeling functions $\ell^{\mathrm{F}} : \mathrm{V}(D) \to \mathrm{F}$ and $\ell^{\mathrm{R}} : \mathrm{V}(D) \to \mathrm{R} \cup \{\emptyset\}$ assign formulas and rule-names. Now for any given locally correct labeled dag $\mathcal{D} = \left\langle D, \ell^{\mathrm{F}}, \ell^{\mathrm{R}}, \ell^{\mathrm{d}} \right\rangle$ we call $\Gamma_{\mathcal{D}} := \left\{ \ell^{\mathrm{F}}(u) : 0 = \overrightarrow{\deg}(u) = \ell^{\mathrm{d}}\left(u, \ell^{\mathrm{F}}(u)\right) \right\}_{u \in \mathrm{V}(D)}$ the set of *open* (or *undischarged*) *assumptions*, in $\mathcal{D}$. Moreover $\mathcal{D} = \left\langle D, \ell^{\mathrm{F}}, \ell^{\mathrm{R}}, \ell^{\mathrm{d}} \right\rangle$ is called an encoded *dag-like natural deduction of* $\ell^{\mathrm{F}}(root(D))$ *from* $\Gamma_{\mathcal{D}}$. In particular, if $\Gamma_{\mathcal{D}} = \emptyset$, then $\mathcal{D}$ is called an encoded *dag-like proof of* $\ell^{\mathrm{F}}(root(D))$.

Lemma

There is a $1-1$ correspondence between plain and encoded dag-like natural deductions (in particular, proofs).

§2.2. More on local correctness

Definition

Definition

Local correctness conditions for ℓ^{d} are as follows, where
$\ell^{\mathrm{RF}}(u) = (\rightarrow I)_{\alpha}$ stands for
$\ell^{\mathrm{R}}(u) = (\rightarrow I) \wedge \ell^{\mathrm{F}}(u) = \alpha \rightarrow \ell^{\mathrm{F}}\left(u^{(1)}\right)$.

Definition

Local correctness conditions for ℓ^{d} are as follows, where $\ell^{\mathrm{RF}}(u) = (\to I)_\alpha$ stands for $\ell^{\mathrm{R}}(u) = (\to I) \wedge \ell^{\mathrm{F}}(u) = \alpha \to \ell^{\mathrm{F}}\left(u^{(1)}\right)$.

1. If $\overleftarrow{\deg}(u) = 0$, i.e. u is the root, then $\ell^{\mathrm{d}}(u, \alpha) = 1$ iff $\ell^{\mathrm{RF}}(u) = (\to I)_\alpha$.

Definition

Local correctness conditions for ℓ^{d} are as follows, where $\ell^{\mathrm{RF}}(u) = (\rightarrow I)_{\alpha}$ stands for $\ell^{\mathrm{R}}(u) = (\rightarrow I) \wedge \ell^{\mathrm{F}}(u) = \alpha \rightarrow \ell^{\mathrm{F}}\left(u^{(1)}\right)$.

1. If $\overleftarrow{\deg}(u) = 0$, i.e. u is the root, then $\ell^{\mathrm{d}}(u, \alpha) = 1$ iff $\ell^{\mathrm{RF}}(u) = (\rightarrow I)_{\alpha}$.
2. If $\ell^{\mathrm{RF}}(u) = (\rightarrow I)_{\alpha}$, then $\ell^{\mathrm{d}}(u, \alpha) = 1$.

Definition

Local correctness conditions for ℓ^{d} are as follows, where $\ell^{\mathrm{RF}}(u) = (\rightarrow I)_{\alpha}$ stands for $\ell^{\mathrm{R}}(u) = (\rightarrow I) \wedge \ell^{\mathrm{F}}(u) = \alpha \rightarrow \ell^{\mathrm{F}}\left(u^{(1)}\right)$.

1. If $\overleftarrow{\deg}(u) = 0$, i.e. u is the root, then $\ell^{\mathrm{d}}(u, \alpha) = 1$ iff $\ell^{\mathrm{RF}}(u) = (\rightarrow I)_{\alpha}$.
2. If $\ell^{\mathrm{RF}}(u) = (\rightarrow I)_{\alpha}$, then $\ell^{\mathrm{d}}(u, \alpha) = 1$.
3. If $\overleftarrow{\deg}(u) > 0$ and $\ell^{\mathrm{RF}}(u) \neq (\rightarrow I)_{\alpha}$, then $\ell^{\mathrm{d}}(u, \alpha) = \prod_{i=1}^{\overleftarrow{\deg}(u)} \ell^{\mathrm{d}}\left(u_{(i)}, \alpha\right)$.