دانشکدگان علوم
دانشکده ریاضی، آمار و علوم کامپیوتر

پیاده سازی قضیه درونیابی یکنواخت در منطق شهودی گزاره‌ای با استفاده از اثبات یار Coq

نگارنده

اشا سروش‌پور

استاد راهنما: دکتر مجید علی‌زاده

پایان‌نامه برای دریافت درجه کارشناسی
در رشته علوم کامپیوتر

امرداد ۱۴۰۲

چکیده

قضیه درونیابی کریگ یکی از قضایای درونیابی برای منطق‌های گوناگون است که برای آن هم اثبات‌های معناشناسانه (جبری، کریپکی و رسته‌ای) وجود دارد و هم اثبات‌های نحوی. اثبات مایهارا یکی از معروف‌ترین اثبات‌های نحوی است که اولین بار برای منطق کلاسیک معرفی شده است. در این پروژه این قضیه را به روش مایهارا برای دو منطق گزاره‌ای کلاسیک و شهودی از رده‌ی منطق های زیرساختی، در اثبات‌یار Coq پیاده‌سازی می‌کنیم.

سپاسگزاری

از استاد راهنمای گرانقدر خود جناب آقای دکتر مجید علی‌زاده و همچنین آقای علیرضا محمودیان بی‌اندازه قدردان و سپاسگزارم چراکه بدون یاری آن‌ها به ثمر رسیدن این پروژه ممکن نبود.
همچنین لازم می‌بینم از آقای دکتر مجتبی مجتهدی نیز تشکرات ویژه خود را به عمل بیاورم که بدون ایمان ایشان به بنده و راهنمایی‌هایشان شاید هیچگاه در این مسیر قدم بر نمی‌داشتم.

پیشگفتار

در ۲۱ ژوئن سال ۱۹۷۶ کنث اپل[1] و ولفگنگ هیکن[2] از دانشگاه ایلینوز اولین کسانی بودند که اعلام کردند پس از سال‌ها تلاش ریاضیدانان توانستند قضیه چهاررنگ [3] را با استفاده از کامپیوتر ثابت کنند. اثبات شامل ۱۸۳۴ حالت مختلف بود که باید توسط کامپیوتر طی هزاران ساعت بررسی می‌شدند و بیش از ۴۰۰ صفحه microfiche که توسط دستیار هیکن، دروثیا بلوستین [4] به صورت دستی بررسی شد. این مورد که اثبات دقیق این قضیه توسط انسان قابل دنبال کردن [5] نبود در آن زمان و حتی تا به الان هم بحث‌برانگیز است. در سال ۱۹۹۶ تیمی دیگر از ریاضیدانان توانستند این قضیه را تنها با بررسی ۶۳۳ حالت اثبات کنند [۱۲] و در سال ۲۰۰۵ بنجامین ورنر و جورج گانتیر اثبات صوری قابل دنبال کردن این قضیه را در اثبات‌یار Coq پیاده‌سازی کردند و در این مسیر کتابخانه SSreflect را طراحی کردند که تا به امروز از آن به عنوان یکی از مهم‌ترین و کاربردی‌ترین کتابخانه‌های Coq یاد می‌شود.
مثالی که زده شد یکی از مصداق‌های بارز و مهم همکاری انسان و کامپیوتر در پیشرفت ریاضیات و بالعکس است. افزایش روزافزون اهمیت اثبات‌یارها و کاربردشان در جهان امروز قابل انکار نیست. در این پروژه سعی شده با صوری سازی [6] یکی از مهم‌ترین قضایای منطق در اثبات‌یار [7] Coq گامی هرچند کوچک در مسیر این همکاری برداشته شود.
در این پایان‌نامه ابتدا به توضیح قضیه درونیابی [8]، اهمیت و صورت کلی آن پرداخته می‌شود سپس توضیحاتی اجمالی درباره اثبات‌یار Coq و ساختار آن ارائه و سپس حساب رشته‌ها [9] که برای اثبات ما مورد نیاز است تعریف می‌شود. در بخش‌های پایانی به توضیح اثبات کامل قضیه درونیابی کریگ به روش مایهارا [10] و توضیح جزئیات تکنیکی مهم در پیاده سازی اثبات این قضیه پرداخته می‌شود.

---

[1] Kenneth Appel
[2] Wolfgang Haken
[3] Four color theorem
[4] Dorothea Blostein
[5] Non-surveyable proof
[6] formalization
[7] proof assistant
[8] interpolation theorem
[9] sequent calculus
[10] Maehara

کد کتابخانه‌های نوشته‌شده و اثبات کامل قضیه در این پیوند قابل مشاهده است. فایل Lang.v تعریف زبان منطق ما است. فایل multiset.v کتابخانه توسعه یافته برای تعریف مجموعه‌های مکرر [11] است. فایل LK.v و LKI.v تعریف حساب رشته‌های منطق گزاره‌ای کلاسیک و منطق گزاره‌ای شهودی همراه با قضایای مرتبط است و در پایان فایل‌های Interpol_theorem.v و Interpol_theorem_intuitionistic.v اثبات کامل قضیه درونیابی را در خود دارد.

[11] multiset

فهرست مطالب

فصل ۱

قضیه‌ی درونیابی کریگ، اهمیت و نتایج آن

۱.۱ صورت کلی قضیه درونیابی

به طور خلاصه قضیه درونیابی در یک منطق فرضی بیان می‌کند که اگر $A \vdash B$ در آن منطق معتبر باشد، گزاره‌ای درونیاب [۱] به اسم H در زبان مشترک A, B وجود دارد که $A \vdash H$ و $H \vdash B$ معتبر است. برای مثال اگر در منطق گزاره‌ای کلاسیک یا شهودی این موضوع را بررسی کنید فرض کنید گزاره $A(p,q)$ حاصل از اتم‌های p و q وجود دارد و همچنین گزاره‌ی دیگری مثل $B(q,r)$ تنها شامل اتم‌های q و r وجود دارد، آنگاه اگر $A \vdash B$ معتبر باشد آیا گزاره‌ای مثل $H(q)$ تنها شامل اتم q وجود دارد که رابطه‌ی $A(p,q) \vdash H(q) \vdash B(q,r)$ صادق باشد ؟
تفسیری که از قضیه درونیابی ارائه شد به قضیه درونیابی کریگ معروف است که توسط ویلیام کریگ [۲] در سال ۱۹۵۷ اولین بار معرفی و اثبات شد [۴]. از دیگر تفسیرهای مرسوم قضیه درونیابی، قضیه درونیابی لیندون [۳] [۱۰] است که بیان می‌کند قطبیت اتم‌های درونیاب با قطبیت اتم‌های گزاره A یکسان است.

تعریف ۱.۱. (*قطبیت رخداد اتم*)
فرض کنید π یک رخداد از یک اتم داخل گزاره‌ی A باشد. آنگاه می‌گوییم
اگر گزاره A اتمی باشد رخداد π مثبت است.
اگر رخداد π در A مثبت (معادلا منفی) باشد آنگاه رخداد π در $A \wedge B, B \wedge A, A \vee B, B \vee$

[1] interpolant
[2] William Craig
[3] Lyndon

$A, B \supset A, \exists x\ A, \forall x\ A$ هم مثبت (معادلا منفی) است و در $A \supset B, \neg A$ منفی (معادلا مثبت) است.

از قضیه‌ی درونیابی برداشت‌های متفاوتی می‌توان داشت برای مثال اگر منطقی دارای عملگرهای وجهی [۴] مثلا $\Box_1, \Box_2, \Box_3$ باشد و داشته باشیم $A(\Box_1, \Box_2) \vdash B(\Box_2, \Box_3)$ آنگاه اگر زبان مشترک درونیاب، عملگرهای وجهی را هم در نظر بگیرد قضیه‌ی درونیابی به وجود گزاره‌ای مثل $H(\Box_2)$ تعبیر می‌شود که $A(\Box_1, \Box_2) \vdash H(\Box_2) \vdash B(\Box_2, \Box_3)$ در منطق وجهی ما صادق است.
به طور عام یکی از مسائلی که قضیه درونیابی درباره آن صحبت می‌کند این است که اگر مثلا گزاره A نتایجی را درباره مقادیر یا ویژگی‌های $\{p, q\}$ بیان کند و این نتایج، استلزام گزاره B درباره مقادیر یا ویژگی‌های $\{q, r\}$ را نتیجه دهد، باید از درون A گزاره‌ای مثلا $H(q)$ استخراج شود که عامل این استلزام است.
از دیگر مسائل مرتبط با قضیه درونیابی قدرت بیان منطق مورد نظر است، ابتدا این را با یک مثال ساده تبیین می‌کنیم. گزاره زیر را در زیرمجموعه‌ای از منطق گزاره‌ای کلاسیک فرض کنید که تنها شامل استلزام [۵] ($\supset$) است.

$$p \supset p \vdash r \supset r$$

درونیاب گزاره بالا $\top$ است که در زبان ما وجود ندارد به طور مشابه اگر

$$p \wedge \neg p \vdash q \wedge \neg q$$

گزاره ما در زبانی شامل $\wedge, \neg$ باشد درونیاب آن $\bot$ است که بازهم قابل دسترسی نیست.
درست است که زبانی که شامل $\bot$ یا $\top$ نباشد خیلی غیرمعمول است اما این مثال ساده‌ای است تا ارتباط مفهوم درونیابی را با شهودی که ما از قدرت بیان یک منطق داریم نشان دهد. مثال دقیق‌تر این موضوع می‌تواند این باشد که قضیه درونیابی در منطق زمانی خطی شکست می‌خورد و دلیل آن این است که عملگر وجهی *تا*[۶]، با گزاره‌های اتمی خاصی قابل تعریف است اما بدون آن خیر.
درونیابی همچنین قضیه مهمی در جست‌وجوی اثبات است. فرض کنید قصد داریم درخت اثبات را برای گزاره P جست‌وجو کنیم، در بسیاری از منطق‌ها حساب رشته‌های معادل آن منطق، شامل قاعده برش به شکل زیر است.

$$\text{cut}\ \frac{\Gamma \Rightarrow A, \Delta \qquad \Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

اگر در منطق ما قضیه *حذف برش* قابل اثبات نباشد، قاعده برش تنها قاعده‌ای است که برخلاف بقیه قواعد استنتاج، فرض آن مستقیما قابل دستیابی نیست. قضیه درونیابی در این شرایط می‌تواند کاندیدهای گزاره‌ی بریده شده A را به ما بدهد.

[4] Modal
[5] implication
[6] Until

فصل ۲

اثبات‌یارها، انواع و ساختار آن‌ها

در علوم کامپیوتر اثبات‌یار به برنامه‌ای گفته می‌شود که انسان را در نوشتن و ساختن اثبات‌های صوری معمولا با محیطی تعاملی یاری می‌دهد. اثبات‌یارها انواع مختلفی دارند و بر پایه نظریات مختلفی ساخته شده‌اند که از معروف‌ترین آن‌ها می‌توان به Agda Prolog، Coq، lean، و Isabelle اشاره کرد. هر کدام از این اثبات‌یارها توانایی و ویژگی‌های خاص خود را دارد. ما برای صوری‌سازی قضیه مدنظر خودمان از اثبات‌یار Coq استفاده می‌کنیم.
اثبات‌یار Coq بر پایه شاخه‌ای از نظریه انواع [۱] به اسم حساب ساخت‌های استقرایی [۲] که خود مشتقی از حساب ساخت‌ها [۳] است بنا نهاده شده‌است. اثبات‌یار Coq به ما اجازه تعریف قضایا و لم‌ها را در قالب نوع [۴] می‌دهد و با ساختن ترم از آن نوع در محیط تعاملی [۵] Coq می‌توانیم آن گزاره معادل نوع را اثبات کنیم. هر نوع، از یک یا چند سازنده [۶] به صورت استقرایی تشکیل می‌شود. برای مثال تعریف نوع boolean به شکل زیر است.

```
Inductive bool : Set := true : bool | false : bool
```

نوع bool زیرمجموعه‌ای از نوع Set (از انواع ابتدایی Coq) دو سازنده دارد که نحوه ساخته شدن ترم‌های آن نوع را معین می‌کند. در این مثال هیچکدام از این دو سازنده، ورودی یا شرط خاصی ندارند. بنابراین هر ترم از نوع bool یا از سازنده true ساخته شده یا از سازنده false. مثالی دیگر از این مورد اعداد طبیعی است.

[1] Type theory
[2] Caculus of inductive constructions
[3] Calculus of constructions
[4] type
[5] Interactive enviroment
[6] constructor

```
Inductive nat : Set :=
| O : nat
| S : nat -> nat.
```

نوع عدد طبیعی (nat) دو سازنده دارد، یکی سازنده O که هیچ شرطی ندارد و هیچ ورودی نمی‌گیرد ولی ترمی از نوع nat را به ما می‌دهد و دیگری سازنده S که یک ترم از نوع nat می‌گیرد و ترمی جدید از نوع nat برای ما می‌سازد. Coq به ما اجازه تعریف انواع چندریخت هم می‌دهد برای مثال تعریف list را نگاه کنید.

```
Inductive list (A : Type) : Type :=
| nil : list A
| cons : A -> list A -> list A.
```

یک لیست از نوع A دو سازنده دارد، اولین سازنده لیست تهی است که هیچ ورودی ندارد و دومی اضافه‌کننده یک ترم از نوع A به لیست از نوع A است. نوع A می‌تواند هر نوعی مثل bool یا nat یا هر نوع دیگری باشد. به این صورت می‌تواند از هر نوعی یک list ساخت.

فصل ۳

حساب رشته‌ها و منطق‌های زیرساختی

حساب رشته‌ها اولین بار در سال ۱۹۳۵ توسط گنتزن [1] [۶] به عنوان گسترشی بر دستگاه استنتاج طبیعی برای دربرگرفتن شکل طبیعی اثبات گزاره‌ها در ذهن، معرفی شد .

تعریف ۱.۳. *در حساب رشته‌ها هر خط از اثبات، یک رشته به فرم*

$$\Gamma \Rightarrow \Delta$$

است که Γ, Δ لیست‌هایی مرتب از گزاره‌ها هستند و به Γ مجموعه زمینه مقدم [2] و به Δ مجموعه زمینه تالی [3] می‌گوییم.
نماد $\Rightarrow$ را نماد رشته [4] می‌نامیم

تعبیر شهودی یک رشته به این صورت است که عطف [5] اعضای مجموعه‌ی زمینه مقدم، استلزام فصل [6] اعضای مجموعه زمینه تالی را نتیجه می‌دهد.

$$\bigwedge_{i=1}^{k} \Gamma \supset \bigvee_{j=1}^{l} \Delta$$

هر رشته از اثبات، توسط قواعد استنتاج حساب رشته‌ها، از رشته‌های قبلی ساخته می‌شود. قواعد حساب رشته‌ها به دو دسته قواعد ساختاری و قواعد گزاره‌ای تقسیم می‌شود. برای مثال لیست زیر نمونه‌ای از قواعد استاندارد حساب رشته‌ها برای منطق گزاره‌ای کلاسیک است.

[1] Gerhard Gentzen
[2] anticedent
[3] succedent
[4] sequent arrow
[5] conjunction
[6] disjunction

$$\frac{\Gamma \Rightarrow \Delta}{\Gamma, p \Rightarrow \Delta}\ \text{lW} \qquad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, p}\ \text{rW}$$

$$\frac{\Gamma, p, p \Rightarrow \Delta}{\Gamma, p \Rightarrow \Delta}\ \text{lC} \qquad \frac{\Gamma \Rightarrow \Delta, p, p}{\Gamma \Rightarrow \Delta, p}\ \text{rC}$$

شکل ۱.۳: قواعد ساختاری حساب رشته‌ها برای منطق گزاره‌ای کلاسیک

$$\frac{}{A \Rightarrow A}\ \text{Axiom}$$

$$\frac{\Gamma \Rightarrow p, \Delta}{\Gamma, \neg p \Rightarrow \Delta}\ \neg l \qquad \frac{\Gamma, p \Rightarrow \Delta}{\Gamma \Rightarrow \neg p, \Delta}\ \neg r$$

$$\frac{\Gamma, b \Rightarrow \Delta}{\Gamma, a \wedge b \Rightarrow \Delta}\ \wedge l2 \qquad \frac{\Gamma, a \Rightarrow \Delta}{\Gamma, a \wedge b \Rightarrow \Delta}\ \wedge l1$$

$$\frac{\Gamma \Rightarrow \Delta, a \qquad \Gamma \Rightarrow \Delta, b}{\Gamma \Rightarrow \Delta, a \wedge b}\ \wedge r$$

$$\frac{\Gamma, a \Rightarrow \Delta \qquad \Gamma, b \Rightarrow \Delta}{\Gamma, a \vee b \Rightarrow \Delta}\ \vee l$$

$$\frac{\Gamma \Rightarrow \Delta, b}{\Gamma \Rightarrow \Delta, a \vee b}\ \vee r2 \qquad \frac{\Gamma \Rightarrow \Delta, a}{\Gamma \Rightarrow \Delta, a \vee b}\ \vee r1$$

$$\frac{\Gamma \Rightarrow \Delta, a \qquad \Gamma, b \Rightarrow \Delta}{\Gamma, a \supset b \Rightarrow \Delta}\ \supset l$$

$$\frac{\Gamma, a \Rightarrow \Delta, b}{\Gamma \Rightarrow \Delta, a \supset b}\ \supset r$$

شکل ۲.۳: قواعد گزاره‌ای حساب رشته‌های منطق گزاره‌ای کلاسیک

برای مثال قضیه modus ponens را در این حساب اثبات می‌کنیم.

قضیه ۲.۳.

$$\vdash p, p \supset q \Rightarrow q$$

اثبات.

$$\dfrac{\dfrac{\text{Axiom}\ \overline{q \Rightarrow q}}{\text{lW}\ \dfrac{}{p, q \Rightarrow q}} \qquad \dfrac{\text{Axiom}\ \overline{p \Rightarrow p}}{\text{rW}\ \dfrac{}{p \Rightarrow q, p}}}{\supset l\ \dfrac{}{p, p \supset q \Rightarrow q}}$$

□

منطق‌های زیرساختی به منطق‌هایی گفته می‌شود که توسط حساب رشته‌ها بدون یک یا چند قواعد ساختاری تولید می‌شوند. در این پروژه تمرکز ما روی منطق گزاره‌ای کلاسیک و منطق گزاره‌ای شهودی از زیرمجموعه منطق‌های زیرساختی است.

فصل ۴

مفاهیم پایه پیاده‌سازی‌شده یا استفاده‌شده

۱.۴ کتابخانه PeanoNat و نوع option

۱.۱.۴ کتابخانه PeanoNat

کتابخانه PeanoNat از مجموعه کتابخانه‌های استاندارد Coq، تنها کتابخانه‌ی ازپیش‌تعریف‌شده‌ای است که استفاده می‌کنیم. باقی کتابخانه‌ها و قضایا و تعاریف مورد نیاز را خودمان باید بنویسیم و اثبات کنیم. در این بخش به توضیح اجمالی کتابخانه PeanoNat که برای تعریف اعداد طبیعی استفاده می‌شود می‌پردازیم.

تعریف ۱.۴. *یک عدد طبیعی از نوع nat یک نوع استقرایی است که دو سازنده دارد. سازنده اول هیچ چیز نمی‌گیرد و نماد عدد طبیعی 0 است و سازنده دوم نماد تالی یک عدد طبیعی است که آن را با Sn نشان می‌دهیم.*

```
Inductive nat : Set :=
| O : nat
| S : nat -> nat.
```

به این شکل که مثلا عدد 1 تالی عدد 0 است پس آن را با SO نمایش می‌دهیم و مثلا عدد 3 را با $SSSO$

تعریف ۲.۴. *جمع دو عدد طبیعی n و m به صورت بازگشتی روی n تعریف می‌شود، به این شکل که اگر $n = O$ باشد حاصل m است وگرنه حاصل $S((n-1)+m)$ است.*

```
Fixpoint add n m :=
match n with
| O => m
| S p => S (p + m)
end
```

تعریف ۳.۴. *تساوی دو عدد طبیعی، یک تابع بازگشتی [1] تصمیم‌پذیر [2] است. برای تساوی دو عدد n و m اگر هر دو صفر باشند مساوی هستند، اگر یکی صفر باشد و دیگری صفر نباشد مساوی نیستند، وگرنه تساوی به حالت $n-1, m-1$ کاهش پیدا می‌کند.*

```
Fixpoint eqb n m : bool :=
match n, m with
  | O, O => true
  | O, S _ => false
  | S _, O => false
  | S n', S m' => eqb n' m'
end.
```

تعریف ۴.۴. *نیای یک عدد طبیعی به این معنی است که یا عدد O است که در آن صورت خود O نیای آن است وگرنه نیای آن $n-1$ است.*

```
Definition pred n :=
match n with
  | O => n
  | S u => u
end.
```

تعریف ۵.۴. *کمینه و بیشینه دو عدد طبیعی m و n به صورت بازگشتی روی n تعریف می‌شود، به این صورت که مثلا برای بیشینه اگر یکی از دو عدد صفر باشد دیگری بیشینه است وگرنه به حالت $n-1, m-1$ کاهش پیدا می‌کند. کمینه هم به صورت مشابه تعریف می‌شود.*

[1] recursive
[2] decidable

```
  Fixpoint max n m :=
  match n, m with
    | 0, _ => m
    | S n', 0 => n
    | S n', S m' => S (max n' m')
  end.

Fixpoint min n m :=
  match n, m with
    | 0, _ => 0
    | S n', 0 => 0
    | S n', S m' => S (min n' m')
  end.
```

تعریف ۶.۴. *کوچکتر بودن یک عدد از دیگری نیز به صورت بازگشتی تعریف می‌شود، به این معنی که عدد O از هر عددی کوچکتر است، وگرنه اگر دیگری O بود خروجی false است در غیر این دو حالت مسئله به $n-1, m-1$ کاهش پیدا می‌کند.*

```
Fixpoint leb n m : bool :=
match n, m with
  | 0, _ => true
  | _, 0 => false
  | S n', S m' => leb n' m'
end.
```

۲.۱.۴ نوع option

نوع option یکی از انوع پیشفرض Coq است که ما از آن برای تعریف قواعد استنتاج برای منطق شهودی استفاده می‌کنیم. option یک نوع چندریخت است که برای تعریف توابع جزئی استفاده می‌شود. این نوع یک نوع ورودی می‌گیرد و دو سازنده دارد، نخستین سازنده None است که به معنی این است که تابع فرضی ما هیچ خروجی ندارد و دومین Some است که یک ترم از نوع ورودی را ورودی می‌گیرد و به معنی خروجی داشتن تابع ما از نوع مدنظر است.

```
Inductive option (A:Type) : Type :=
| Some : A -> option A
| None : option A.
```

۲.۴ زبان

تعریف ۷.۴. *مجموعه اتم‌های زبان ($\mathbb{A}$) شمارا هستند برای همین آن‌ها را از نوع اعداد طبیعی در نظر می‌گیریم.*

```
Definition atom := nat.
```

تعریف ۸.۴. *اجزای زبان ما شامل $\bot$، فصل $\vee$، عطف $\wedge$ و استلزام $\supset$ می‌باشد. تعریف استقرایی مجموعه گزاره‌های زبان ما به شکل زیر است:*

$$\mathbb{P} := \{\mathbb{A}|\bot|\mathbb{P} \vee \mathbb{P}|\mathbb{P} \wedge \mathbb{P}|\mathbb{P} \supset \mathbb{P}\}$$

در کد، مجموعه گزاره‌های خود را یک نوع استقرایی به صورت زیر تعریف می‌کنیم.

```
Inductive prop : Type :=
| Var : atom  -> prop
| Bot : prop
| Disj : prop -> prop -> prop
| Conj : prop -> prop -> prop
| LDiv : prop -> prop -> prop.
```

همچنین $\top$ را $\bot \supset \bot$ تعریف می‌کنیم.

```
Definition Top := LDiv Bot Bot.
```

و نقیض یک گزاره $(\neg p)$ را $p \supset \bot$ تعریف می‌کنیم.

```
Definition Neg (p : prop) : prop := LDiv p Bot.
```

تعریف ۹.۴. می‌دانیم که مجموعه گزاره‌های ما شمارا است. ما این مسئله را به عنوان یک اصل در دستگاه خود فرض می‌کنیم. به این صورت که دو تابع از گزاره‌ها به اعداد طبیعی $(prop\_to\_nat)$ و از اعداد طبیعی به گزاره‌ها (nat_to_prop) وجود دارد که ترکیبشان باهم تابع همانی می‌شود.

```
Axiom prop_to_nat: prop -> nat.
Axiom nat_to_prop: nat -> prop.

Axiom prop_to_prop: forall p, nat_to_prop(prop_to_nat(p)) = p.
Axiom nat_to_nat: forall n, prop_to_nat(nat_to_prop(n)) = n.
```

ما از نمادگذاری های زیر در کد استفاده می‌کنیم:

نماد	تابع یا تعریف
$\hat{}x\_$	*Var*
$\wedge$	*Conjunction*
$\vee$	*Disjunction*
$\supset$	*implication*
$\neg$	*Neg*
$\top$	*Top*
$\bot$	*Bot*

```
Notation ^x_ x := (Var x) (at level 30).
Notation ⊥ := Bot.
Notation ⊤ := Top.
Notation ¬P := (Neg P) (at level 31).
Infix ∧ := Conj (left associativity, at level 32).
Infix ∨ := Disj (left associativity, at level 33).
Infix ⊃ := LDiv (right associativity, at level 34).
```

۳.۴ مجموعه مکرر

ما برای تعریف مجموعه‌های زمینه مقدم و تالی، از مجموعه‌ی مکرر[3] استفاده می‌کنیم. مجموعه مکرر مجموعه‌ای است که تکرار اعضا در آن مجاز است برای مثال مجموعه مکرر $\{1,2\}$ با مجموعه مکرر $\{1,2,1\}$ متفاوت است. همچنین ما برای مسائل تکنیکی و سادگی کار مجموعه‌های مکرر خود را نامتناهی فرض می‌کنیم. این کار با اینکه قدرت استفاده از قضیه استقرا را از ما می‌گیرد ولی به سادگی اثبات بعضی قضایای مورد نیاز کمک شایانی می‌کند. در اکثر کتب معمولا مجموعه‌های زمینه حساب رشته‌ها را لیست فرض می‌کنند این موضوع دو مشکل اساسی برای پیاده‌سازی ما ایجاد می‌کند.

۱. اول اینکه در صورت فرض کردن مجموعه‌های زمینه به شکل لیست باید قواعد جابه‌جایی [4] را نیز به قواعد دستگاه استنتاجی خود اضافه کنیم. در روش مایهارا اثبات درونیابی برای این قواعد کار بسیار پیچیده‌ای است.

۲. ثانیا این کار ما را نیازمند قضایای بدیهی بیشتر و پیچیده‌تری می‌کند. یک ریاضیدان در اثبات ریاضی خود می‌تواند جابه‌جایی گزاره‌های لیست را با قاعده جابه‌جایی تا حد دلخواه بدیهی فرض کند و با گذاشتن یک (...) کار را فیصله دهد ولی در اثبات‌یار تمامی گام‌های اثبات باید دقیق نوشته شوند مثلا این یک فرض بدیهی نیست که اگر یک عدد در لیست حاصل از چسباندن دو لیست باشد آنگاه در یکی از آنها وجود دارد، یا قضایای مشابه مثل شکستن یک لیست به دو لیست و پیدا کردن حاصل آن.

شاید تصور شود که استفاده از مجموعه‌های متناهی کار درست‌تری است. وقتی قواعد استنتاج ما استقرایی هستند عملا اجازه تولید مجموعه زمینه نامتناهی را به ما نمی‌دهد، اما این کار نیز پیچیدگی‌هایی غیرضروری را وارد پیاده‌سازی می‌کند. برای تعریف یک نوع متناهی راه‌های متفاوتی وجود دارد که بدیهی‌ترین آن‌ها این است که آن را استقرایی تعریف کنیم، به طوری که باید بتوانیم در این مثال به تک‌تک اعضای مجموعه دسترسی داشته باشیم. این کار برای مجموعه‌ها پیچیدگی‌های زیادی اضافه می‌کند برای مثال برای خروجی دادن تکرار یک عضو در مجموعه باید تماما آن را پیمایش کنیم؛ یا کم کردن یک عضو از یک مجموعه بسته به تعریف ما می‌تواند کار دشواری باشد، مثلا اگر اعضای یکسان را کنار هم نگهداری کنیم اضافه کردن عضو کار پیچیده‌ای می‌شود و اگر اعضا را به طور درهم نگهداری کنیم اینکه کدام عضو را پاک کنیم حالات زیادی به مسئله اضافه می‌کند.

---

[3] multiset

[4] exchange rules

۱.۳.۴ تعاریف

تعریف ۱۰.۴. *ما یک مجموعه مکرر را نوعی از توابع از اعداد طبیعی به اعداد طبیعی تعریف می‌کنیم معنای این تعریف این است که خروجی تابع مدنظر برای هر عدد طبیعی تعداد تکرار آن عدد در مجموعه را به ما نشان می‌دهد.*

```
Definition U := nat .
Definition multiset := U -> nat.
```

از اینجا به بعد، به تعریف روابط بین مجموعه‌های مکرر می‌پردازیم.

تعریف ۱۱.۴. *عضویت عدد طبیعی x در مجموعه A به معنی قضیه‌ای است که نشان می‌دهد خروجی تابع A برای عدد x بزرگتر از یک است.*

```
Definition In (x : U) (A : multiset)  : Prop := 1 <= A x.
```

تعریف ۱۲.۴. *اینکه مجموعه مکرر A زیرمجموعه مجموعه مکرر B باشد به معنی قضیه‌ای است که بیان می‌کند:*

$$\forall x, A(x) \leq B(x)$$

```
Definition Included (A B : multiset) : Prop :=
forall x, A x <= B x.
```

تعریف ۱۳.۴. *اجتماع دو مجموعه مکرر A و B را تابعی تعریف می‌کنیم که برای هر x دلخواه جمع دو تابع مذکور را خروجی می‌دهد.*

```
Definition Union (A B : multiset) : multiset :=
fun (x : U) => (A x) + (B x).
```

تعریف ما با تعریف سنتی اجتماع در مجموعه‌های مکرر کمی متفاوت است. در مجموعه‌های مکرر اجتماع، بیشینه تکرار در دو مجموعه تعریف می‌شود و تعریفی که ما ارائه دادیم تعریف Sum در مجموعه‌های مکرر است. دلیل این مسئله این است که ما عملا هیچگاه به تعریف سنتی اجتماع نیازی پیدا نخواهیم کرد.

تعریف ۱۴.۴. *اشتراک دو مجموعه A و B به معنی تابعی است که برای هر x ورودی کمینه‌ی خروجی این دو تابع را خروجی می‌دهد.*

```
Definition Intersection (A B : multiset) : multiset :=
fun (x : U) => min (A x) (B x).
```

تعریف ۱۵.۴. *اضافه کردن عضو x به مجموعه A به معنی تابعی است که برای هر عدد y به جز x، $A(y)$ را خروجی می‌دهد و برای x، $A(x)+1$ را خروجی می‌دهد.*

```
Definition SAdd (A : multiset) (x : U) : multiset :=
fun (x' : U) => match ( x =? x') with
    | true => S (A x')
    |false => A x'
end.
```

تعریف ۱۶.۴. *کم کردن عضو x به مجموعه A به معنی تابعی است که برای هر عدد y به جز x، $A(y)$ را خروجی می‌دهد و برای x، $predA(x)$ را خروجی می‌دهد.*

```
Definition Remove (A : multiset) (x : U) : multiset :=
fun (x' : U) => match ( x =? x') with
    | true => pred (A x')
    |false => A x'
end.
```

تعریف ۱۷.۴. *کم کردن مجموعه A از مجموعه B به معنی تابعی است که برای هر عدد تفاضل دو مجموعه مذکور را خروجی می‌دهد.*

```
Definition Diff (A B : multiset) : multiset :=
fun (x : U) => (A x) - (B x).
```

تعریف ۱۸.۴. *مجموعه تهی تابعی است که همواره 0 خروجی می‌دهد.*

```
Definition EmptySet : multiset := fun (x : U) => 0.
```

تعریف ۱۹.۴. *مجموعه تک عضوی $\{x\}$ تابعی است که برای x عدد 1 را خروجی می‌دهد و برای هر عدد دیگری 0 را خروجی می‌دهد.*

```
Definition Singleton (x:U) : multiset :=
fun (x' : U) => match ( x =? x') with
    | true => 1
    |false => 0
end.
```

تعریف ۲۰.۴. *مساوی بودن دو مجموعه به معنی قضیه‌ای است که نشان می‌دهد برای هر عددی خروجی دو تابع مذکور یکسان است.*

```
Definition Equal (A B : multiset) : Prop :=
forall (x : U), (A x) = (B x).
```

برای استفاده از این تعریف باید این اصل هم به نظریه اضافه کنیم که تعریفی که ما از تساوی دادیم مساوی بودن را هم نتیجه می‌دهد زیرا در عمل هر دو مجموعه ترم‌های متفاوتی هستند حتی اگر خروجی‌شان همواره یکسان باشد.

```
Axiom Extensionality_multiset : forall A B, A == B -> A = B.
```

تعریف ۲۱.۴. *تنها دو تعریف دیگر نیاز داریم که اولی، تابعی برای تبدیل نوع option prop به نوع مجموعه مکرر است. این تعریف را در اثبات قضیه برای منطق شهودی استفاده خواهیم کرد.*

```
Definition option_to_multiset (op : option prop) :=
    match op with
        | None =>  ∅
```

```
        | Some p => {{ p }}
    end.
```

تعریف ۲۲.۴. *و دومی، برای اضافه کردن یک ترم از نوع option prop به یک مجموعه مکرر است.*

```
Definition option_add (s : multiset) (op : option prop) : multiset :=
    match op with
        | None => s
        | Some p => s  ⨥  p
    end.
```

ما از نمادگذاری های زیر در کد استفاده می‌کنیم:

تابع یا تعریف	نماد
Union	$\cup$
Intersection	$\cap$
Diff	$\setminus$
In	$\in$
Included	$\subset$
Equal	==
SAdd	$\hat{+}$
Remove	$\cdot-$
Singleton	{{ }}

```
Notation  ∅  := EmptySet.
Infix ∪:= Union (left associativity, at level 62).
Infix ∩ := Intersection (left associativity, at level 62).
Infix \ := Diff (left associativity, at level 63).
Infix ∈ := In (at level 60).
Infix ⊆ := Included (at level 71).
Infix == := Equal (at level 70).
Infix ⨥ := SAdd (at level 64).
```

```
Infix ·— := Remove (at level 65).
Notation {{ n }} := (Singleton n).
```

۲.۳.۴ قضایای اثبات‌شده

قضیه‌های اثبات‌شده بسیار زیاد هستند، به همین دلیل اثبات کامل همه آنها را نمی‌آوریم.

لم ۲۳.۴ (emp_minimum).

$$\forall s, \emptyset \subseteq s$$

اثبات. با باز کردن تعریف Included و استفاده از قضیه Nat.le_0_r در کتابخانه PeanoNat این قضیه به‌راحتی اثبات می‌شود.

```
Lemma emp_min : forall s, s ⊆ ∅ -> ∅ = s.
Proof.
    intros. apply Extensionality_multiset.
    unfold Included in H. unfold EmptySet in H.
    assert (forall x : U, s x = 0).
        - intros. rewrite <- Nat.le_0_r. auto.
        - unfold Equal. intros. unfold EmptySet. auto.
Qed.
```

□

لم ۲۴.۴ (incl_reflexive).

$$\forall s, s \subseteq s$$

لم ۲۵.۴ (incl_transitive).

$$\forall s_1\ s_2\ s_3, s_1 \subseteq s_2 \wedge s_2 \subseteq s_3 \rightarrow s_1 \subseteq s_3$$

اثبات. با استفاده از قضیه Nat.le_trans در کتابخانه PeanoNat و باز کردن تعاریف، این قضیه هم به‌سادگی اثبات می‌شود.

```
Lemma incl_transitive : forall s1 s2 s3, s1 ⊆ s2 -> s2 ⊆ s3
-> s1 ⊆ s3.
```

```
Proof.
    unfold ⊆. intros.
    apply (Nat.le_trans (s1 x) (s2 x) (s3 x)); auto.
Qed.
```

□

لم ۲۶.۴ (extensionality).

$$\forall s_1\ s_2, s_1 \subseteq s_2 \to s_2 \subseteq s_1 \to s_1 = s_2$$

لم ۲۷.۴ (extensionality_iff).

$$\forall s_1\ s_2, (s_1 \subseteq s_2 \land s_2 \subseteq s_1) \leftrightarrow s_1 = s_2$$

اثبات. این قضیه با استفاده از لم قبلی و اصل Extensionality_multiset اثبات می‌شود.

```
Lemma extensionality_iff : forall s1 s2, (s1 ⊆ s2 /\ s2 ⊆ s1) <-> s1 = s2.
Proof.
    split; intros; unfold ⊆ in *; try split.
        - apply Extensionality_multiset; unfold ==.
        destruct H. intros. apply Nat.le_antisymm; auto.
        - intros. rewrite H. apply Nat.le_refl.
        - intros. rewrite H. apply Nat.le_refl.
Qed.
```

□

لم ۲۸.۴ (mem_incl).

$$\forall s_1\ s_2, s_1 \subseteq s_2 \to (\forall n, n \in s_1 \to n \in s_2)$$

لم ۲۹.۴ (union_idl).

$$\forall s, \emptyset \cup s = s$$

لم ۳۰.۴ (union_idr).

$$\forall s, s \cup \emptyset = s$$

لم ۳۱.۴ (intersection_empl).

$$\forall s, s \cap \emptyset = \emptyset$$

لم ۳۲.۴ (intersection_empr).

$$\forall s, \emptyset \cap s = \emptyset$$

لم ۳۳.۴ (single_add).

$$\forall a\ b, \{\{a\}\} \hat{+} b = \{\{b\}\} \hat{+} a$$

اثبات. این قضیه با باز کردن تعاریف و حالت‌بندی روی برابری ورودی با a یا b اثبات می‌شود. در اثبات این قضیه از تصمیم‌پذیری برابری نوع اعداد طبیعی و قضیه Nat.eqb_sym استفاده شده است.

```
Lemma single_add: forall a b, {{ a }} ⨥ b = {{ b }} ⨥ a.
Proof.
    intros; apply Extensionality_multiset; unfold ==. unfold ⨥.
    unfold Singleton. intros.
    replace (b =? a) with (a =? b).
    destruct (a =? x) eqn: E; destruct (b =? x) eqn: E'; simpl;try reflexivity.
        - rewrite Nat.eqb_sym; reflexivity.
Qed.
```

□

لم ۳۴.۴ (le_one_cases).

$$\forall n\ m, 1 \leq n + m \to 1 \leq n \vee 1 \leq m$$

اثبات. این لم در اثبات بسیاری از لم‌ها و قضایای مرتبط با عضویت، اهمیت زیادی دارد. این لم را با استقرا روی n ثابت می‌کنیم. حالت پایه که بدیهی است. برای گام استقرا، ما در فرض استقرا داریم که

$$\forall n\ m, 1 \leq n + m \to 1 \leq n \vee 1 \leq m$$

و باید همین قضیه را برای Sn اثبات کنیم. از فرض به نتیجه می‌رسیم.

$$1 \leq Sn + m \to 0 \leq n + m$$

بنابراین

$$0 \leq n \vee 0 \leq m$$

با حالت‌بندی روی این نتیجه و باز کردن تعاریف، گام استقرا اثبات می‌شود.

```
Lemma le_one_cases: forall n m, 1 <= n + m -> 1 <= n \/ 1 <= m.
Proof.
    intros. induction n.
        - auto.
        - simpl in H. rewrite <- Nat.succ_le_mono in H.
        specialize (Nat.add_nonneg_cases n m H) as H'.
        destruct H'.
            + left; rewrite <- Nat.succ_le_mono; assumption.
            + destruct m.
                * left. rewrite <- Nat.succ_le_mono.
                rewrite plus_n_O. assumption.
                * assert (1 <= n + S m).
                {
                    rewrite Nat.add_succ_r. rewrite <- Nat.succ_le_mono.
                    apply le_0_n.
                }
                {
                    apply IHn in H1. destruct H1.
                    - left. apply Nat.le_le_succ_r. assumption.
                    - right. assumption.
                }
Qed.
```

□

لم ۳۵.۴ (le_add_r۲).

$$\forall a\ b\ c, a \leq b \to a \leq b + c$$

لم ۳۶.۴ (union_mem).

$$\forall s_1\ s_2\ x, x \in (s_1 \cup s_2) \leftrightarrow (x \in s_1 \vee x \in s_2)$$

لم ۳۷.۴ (union_comm).

$$\forall s_1\ s_2, s_1 \cup s_2 = s_2 \cup s_1$$

لم ۳۸.۴ (inter_comm).

$$\forall s_1\ s_2, s_1 \cap s_2 = s_2 \cap s_1$$

لم ۳۹.۴ (equal_ext).

$$\forall A\ B, A = B \leftrightarrow A == B$$

لم ۴۰.۴ (add_eq_one).

$$\forall m\ n, m + n = 1 \to m = 1 \vee n = 1$$

لم ۴۱.۴ (add_eq_o).

$$\forall m\ n, m + n = 0 \to m = 0 \wedge n = 0$$

از لم ۴۲.۴ تا لم ۵۰.۴ لم‌های اساسی و مهمی هستند که مکررا در اثبات دقیق درونیابی استفاده می‌شوند.

لم ۴۲.۴ (one_in_union).

$$\forall x\ s_1\ s_2, \{\{x\}\} = s_1 \cup s_2 \to (\{\{x\}\} = s_1 \vee \{\{x\}\} = s_2)$$

اثبات. این لم را برای اثبات راحت‌تر لم بعدی استفاده می‌کنیم. فرض قضیه به ما می‌گوید که $s_1 \cap s_2$ تابعی است که فقط برای x خروجی ۱ می‌دهد و برای بقیه ورودی‌ها خروجی ۰ بنابراین اگر روی مساوی بودن ورودی تابع با x حالت‌بندی کنیم با لم‌هایی که قبلا اثبات کردیم مثل لم ۴۰.۴ به‌راحتی این قضیه اثبات می‌شود.

```
Lemma one_in_union: forall x s1 s2, {{x}} = s1 ∪ s2
-> ({{x}} = s1) \/ {{x}} = s2.
Proof.
    intros. rewrite equal_ext in H. unfold == in H. unfold ∪ in H.
    specialize (H x) as H'. unfold Singleton in H'.
    rewrite PeanoNat.Nat.eqb_refl in H'.
    symmetry in H'. apply add_eq_one in H'. destruct H'.
        - left. rewrite equal_ext. unfold Equal. intros. unfold Singleton.
        destruct (PeanoNat.Nat.eqb x x0) eqn: E.
            + apply PeanoNat.Nat.eqb_eq in E. subst. symmetry. assumption.
            + specialize (H x0) as H'. unfold Singleton in H'. rewrite E in H'.
```

```
        symmetry in H'; apply add_eq_o in H'; destruct H'; auto.
    - right. rewrite equal_ext. unfold Equal. intros. unfold Singleton.
    destruct (PeanoNat.Nat.eqb x x0) eqn: E.
        + apply PeanoNat.Nat.eqb_eq in E. subst. symmetry. assumption.
        + specialize (H x0) as H'. unfold Singleton in H'. rewrite E in H'.
        symmetry in H'; apply add_eq_o in H'; destruct H'; auto.
Qed.
```

□

لم ۴۳.۴ (one_in_union۲).

$$\forall x\ s_1\ s_2, \{\{x\}\} = s_1 \cup s_2 \to ((\{\{x\}\} = s_1 \wedge s_2 = \emptyset) \vee (\{\{x\}\} = s_2 \wedge s_1 = \emptyset))$$

اثبات. روی حالت‌های لم ۴۲.۴ حالت‌بندی می‌کنیم و سپس با حالت‌بندی روی ورودی توابع و حاصل جمعشان با قضایای اثبات‌شده در کتابخانه PeanoNat این قضیه مهم برای کار ما اثبات می‌شود.

```
Lemma one_in_union2: forall x s1 s2, {{x}} = s1 ∪ s2
-> ({{x}} = s1 /\ s2 =  ∅ ) \/ ({{x}} = s2 /\ s1 =  ∅ ).
Proof.
    intros. apply one_in_union in H as H'. rewrite equal_ext in H.
    unfold == in H. unfold ∪ in H.
    destruct H'.
        - left; split; try assumption.
        rewrite equal_ext. unfold Equal. intros. unfold  ∅ .
        destruct (PeanoNat.Nat.eqb x x0) eqn: E.
            +apply PeanoNat.Nat.eqb_eq in E. subst.  specialize (H x0) as H'.
            unfold Singleton in H'; rewrite PeanoNat.Nat.eqb_refl in H';
            inversion H'. reflexivity.
            +specialize (H x0) as H'. unfold Singleton in H'. rewrite E in H'.
            symmetry in H'. apply PeanoNat.Nat.eq_add_0 in H'.
            destruct H'. assumption.
        - right; split; try assumption.
        rewrite equal_ext. unfold Equal. intros. unfold  ∅ .
        destruct (PeanoNat.Nat.eqb x x0) eqn: E.
            +apply PeanoNat.Nat.eqb_eq in E. subst.  specialize (H x0) as H'.
```

```
            unfold Singleton in H'; rewrite PeanoNat.Nat.eqb_refl in H'.
            rewrite <- PeanoNat.Nat.add_succ_comm in H'. inversion H'.
            rewrite PeanoNat.Nat.add_0_r in H1. auto.
            +specialize (H x0) as H'. unfold Singleton in H'. rewrite E in H'.
            symmetry in H'. apply PeanoNat.Nat.eq_add_0 in H'.
            destruct H'. assumption.
Qed.
```

□

لم ۴۴.۴ (add_union_singl).

$$\{\{x\}\} \cup s = s \hat{+} x$$

اثبات. از این قضیه‌ی مهم برای تبدیل SAdd و Union به یکدیگر مکررا استفاده می‌شود. چون قواعد دستگاه استنتاج ما برای راحتی با SAdd نوشته شده‌اند.

```
Lemma add_union_singl: forall s x,
{{x}} ∪ s = s ⨥ x.
Proof.
    intros. apply Extensionality_multiset. unfold ==.
    intros. unfold ∪; unfold Singleton; unfold ⨥ .
    destruct (PeanoNat.Nat.eqb x x0); simpl; reflexivity.
Qed.
```

□

لم ۴۵.۴ (emp_add).

$$\forall x, \emptyset \hat{+} x = \{\{x\}\}$$

لم ۴۶.۴ (ms_add_comm).

$$\forall s\ a\ b, (s \hat{+} a) \hat{+} b = (s \hat{+} b) \hat{+} a$$

لم ۴۷.۴ (add_sing_refl).

$$\forall p\ q, \{\{p\}\} \hat{+} q = \{\{q\}\} \hat{+} p$$

چهار قضیه زیر اهمیت زیادی در کاهش گام استقرا به فرض استقرا در اثبات درونیابی دارند.

لم ۴۸.۴ (add_in_union).

$$\forall s_1\ s_2\ s_3\ p, s_1 \hat{+} p = s_2 \cup s_3 \to (p \in s_2 \vee p \in s_3)$$

اثبات. می‌دانیم که بنا به اصل Extensionality_multiset فرض لم به ما این نتیجه را می‌دهد که برای p حاصل‌جمع دو تابع s_2 و s_3 از یک بزرگتر است. با استفاده از لم ۳۴.۴ و حالت‌بندی روی نتیجه آن این لم اثبات می‌شود.

```
Lemma add_in_union: forall s1 s2 s3 p, s1 ⨥ p = s2 ∪ s3 -> p ∈ s2 \/ p ∈ s3.
Proof. intros.
    rewrite equal_ext in H. unfold Equal in H. specialize (H p) as H'.
    unfold ⨥ in H'. rewrite PeanoNat.Nat.eqb_refl in H'.
    unfold ∪ in H'. assert (1 <= S (s1 p)).
        * apply le_n_S. apply le_0_n.
        * rewrite H' in H0. apply le_one_cases in H0. destruct H0.
            **left. assumption.
            **right. assumption.
Qed.
```

□

لم ۴۹.۴ (in_add_unfold).

$$\forall s_1\ p, p \in s_1 \to (\exists s', s' \hat{+} p = s_1)$$

اثبات. تابع خروجی مدنظر ما $s' = s_1 \cdot - p$ است. ولی اثبات این لم چندان راحت نیست زیرا می‌دانیم که $pred(0) = 0$ است. بنابراین باید از فرض لم استفاده کنیم و نشان دهیم که $1 \leq s_1(p)$ است.

```
Lemma in_add_unfold: forall s1 p, p ∈ s1 -> exists s', s' ⨥ p = s1.
Proof.
    intros. exists (s1 ·- p). rewrite equal_ext. unfold Equal.
    intros. unfold ·- . unfold ⨥ .
    destruct (PeanoNat.Nat.eqb p x) eqn: E.
```

```
    - apply PeanoNat.Nat.eqb_eq in E. subst. unfold In in H.
    apply (PeanoNat.Nat.lt_succ_pred 0). assumption.
    - reflexivity.
Qed.
```

□

لم ۵۰.۴ (add_union_remove).

$$\forall s_1\ s_2\ p, s_1 \hat{+} p = s_2 \hat{+} p \to s_1 = s_2$$

اثبات. این لم با حالت‌بندی روی ورودی تابع و برابری آن با p اثبات می‌شود.

```
Lemma add_union_remove: forall s1 s2 p, s1 ∔ p = s2 ∔ p -> s1 = s2.
Proof.
    intros. rewrite equal_ext. unfold Equal. intros.
    rewrite equal_ext in H. unfold Equal in H. specialize (H x) as H'.
    unfold ∔ in H'.
    destruct (PeanoNat.Nat.eqb p x); inversion H'; subst; reflexivity.
Qed.
```

□

لم ۵۱.۴ (add_union_extrac).

$$\forall s_1\ s_2\ p, s_1 \cup (s_2 \hat{+} p) = (s_1 \cup s_2) \hat{+} p$$

لم ۵۲.۴ (le_S_one).

$$\forall n, 1 \leq Sn$$

لم ۵۳.۴ (in_add).

$$\forall G\ a\ p, a \in G \to a \in (G \hat{+} p)$$

لم ۵۴.۴ (In_emp_false).

$$\forall x, x \in \emptyset \to False$$

۴.۴ فرمول‌ها، توابع و قضیه‌های مرتبط

۱.۴.۴ تعاریف

ما برای اثبات و نشان دادن شرط سوم قضیه‌ی درونیابی به سه تعریف اساسی نیاز داریم.

تعریف ۵۵.۴ (atoms_of)**.** *اولین تعریف ما یک تابع بازگشتی است که بیان می‌کند آیا یک اتم در یک گزاره رخداد دارد یا نه. این تابع یک گزاره از نوع prop و یک atom می‌گیرد و به شکل بازگشتی رخداد آن اتم را به نوع bool خروجی می‌دهد.*

```
Fixpoint atoms_of (p : prop) (a: atom) : bool :=
match p with
    | ^x_a' => if (Nat.eqb a a') then true else false
    | p1 ∧ p2 => (atoms_of p1 a) || (atoms_of p2 a)
    | p1 ∨ p2 => (atoms_of p1 a) || (atoms_of p2 a)
    | p1 ⊃ p2 => (atoms_of p1 a) || (atoms_of p2 a)
    | _ => false
end.
```

تعریف ۵۶.۴ (atom_in)**.** *دومین تعریف ما بیان رخداد یک اتم در یک مجموعه مکرر (از گزاره‌ها) است. این تعریف به معنی گزاره‌ای است که بیان می‌کند، گزاره‌ای وجود دارد که در این مجموعه مکرر عضو است و تابع ۵۵.۴ برای آن خروجی true می‌دهد.*

```
Definition atom_in (s: multiset) (a: atom) : Prop :=
exists (p: prop), (prop_to_nat p) ∈ s /\ atoms_of p a.
```

تعریف ۵۷.۴ (atoms_incl)**.** *سومین تعریف مورد نیاز ما بیان این موضوع است که اتم‌های یک گزاره زیرمجموعه زبان مشترک دو مجموعه مکرر از گزاره‌ها است. این را به گزاره‌ای تعبیر می‌کنیم که بیان می‌کند برای هر اتم اگر آن اتم بر اساس ۵۵.۴ داخل گزاره رخداد داشته باشد آنگاه آن اتم حتما داخل هر دو مجموعه مکرر بنا به تعریف ۵۶.۴ رخداد دارد.*

```
Definition atoms_incl (p: prop) (s1 s2: multiset) : Prop :=
forall (a: atom), (atoms_of p a) ->
(atom_in s1 a) /\ (atom_in s2 a).
```

۲.۴.۴ لم‌ها و قضایا

چون تعداد قضایا و جزئیات اثبات زیاد است به بیان آن‌ها اکتفا می‌کنیم و از اثبات قضایای مشابه پرهیز می‌کنیم.

لم ۵۸.۴ (atomin_add).

$$\forall s\ a\ p, atom\_in(s, a) \to atom\_in(s \hat{+} p, a)$$

اثبات. فرض لم به ما این نتیجه را می‌دهد که p' وجود دارد به طوری که $p' \in s$ و $atoms\_of(p', a)$ کافیست همین p' را به عنوان شاهد به گزاره نتیجه بدهیم و دو شرط آن را اثبات کنیم.

```
Lemma atomin_add: forall s a p, atom_in s a -> atom_in (s ∔̂ p) a.
Proof.
    intros. unfold atom_in. unfold atom_in in H. destruct H as [p' [H H']].
    exists p'. split.
    - unfold In; unfold ∔̂. destruct (PeanoNat.Nat.eqb p (prop_to_nat p')).
      + apply le_S_one.
      + assumption.
    - assumption.
Qed.
```

□

لم ۵۹.۴ (atomin_add_double).

$$\forall s\ a\ p, atom\_in((s \hat{+} p) \hat{+} p, a) \to atom\_in(s \hat{+} p, a)$$

لم ۶۰.۴ (atomin_andr).

$$\forall s\ a\ p\ q, atom\_in(s \hat{+} p, a) \to atom\_in(s \hat{+} (p \wedge q), a)$$

اثبات. فرض لم به ما این نتیجه را می‌دهد که p' وجود دارد به طوری که $p' \in (s \hat{+} p)$ و $atoms\_of(p', a)$ کافیست حال دو حالت وجود دارد یا $p' \in s$ است یا $p' = p$ است. برای حالت اول خود p' را به عنوان شاهد برای گزاره وجودی نتیجه می‌دهیم و برای حالت دوم $p \wedge q$ را.

```
Lemma atomin_andr: forall s a (p q : prop), atom_in (s +̂ p) a
-> atom_in (s +̂ (p ∧ q)) a.
Proof.
    intros. unfold atom_in. unfold atom_in in H. destruct H as [p' [H H']].
    unfold In; unfold +̂. unfold In in H; unfold +̂ in H.
    destruct (PeanoNat.Nat.eqb p (prop_to_nat p')) eqn: E.
        - exists (p ∧ q). split.
            + rewrite PeanoNat.Nat.eqb_refl. apply le_S_one.
            + unfold atoms_of. apply eqb_eq in E. rewrite prop_to_prop in E.
            rewrite prop_to_prop in E. rewrite <- E in H'. rewrite H'.
            rewrite Bool.orb_true_l. reflexivity.
        - exists p'.
        destruct (PeanoNat.Nat.eqb (p ∧ q) (prop_to_nat p')) eqn: E'.
            + apply eqb_eq in E'. rewrite prop_to_prop in E'.
            rewrite prop_to_prop in E'. split.
                * apply le_S_one.
                * assumption.
            + split; assumption.
Qed.
```

□

لم ۶۱.۴ (atomin_andl).

$$\forall s\ a\ p\ q, atom\_in(s\hat{+}q, a) \to atom\_in(s\hat{+}(p \wedge q), a)$$

لم ۶۲.۴ (atomin_orr).

$$\forall s\ a\ p\ q, atom\_in(s\hat{+}p, a) \to atom\_in(s\hat{+}(p \vee q), a)$$

لم ۶۳.۴ (atomin_orl).

$$\forall s\ a\ p\ q, atom\_in(s\hat{+}q, a) \to atom\_in(s\hat{+}(p \vee q), a)$$

لم ۶۴.۴ (atomin_imp).

$$\forall s_1\ s_2\ a\ p\ q, atom\_in((s\hat{+}p) \cup (s\hat{+}q), a) \to atom\_in(s\hat{+}(p \supset q), a)$$

اثبات. بنا به لم ۳۶.۴ p' یا در سمت چپ اجتماع عضویت دارد یا در سمت راست. در دو حالت، یا p' عضو مجموعه اصلی یا برابر با عضو اضافه‌شده است. پس در کل چهار حالت داریم که مشابه قبل اثبات خواهند شد.

```
Lemma atomin_imp: forall s1 s2 a (p q : prop),
atom_in ((s1 ∔̂ p) ∪ (s2 ∔̂ q)) a
-> atom_in (s1 ∪ (s2 ∔̂ p ⊃ q)) a.
Proof.
    intros. unfold atom_in. unfold atom_in in H. destruct H as [p' [H H']].
    rewrite union_mem in H. destruct H; unfold In in H.
        - destruct (PeanoNat.Nat.eqb p (prop_to_nat p')) eqn: E.
            + exists (p ⊃ q); split.
                *unfold In. rewrite add_union_extrac.
                unfold ∔̂.
                rewrite PeanoNat.Nat.eqb_refl. apply le_S_one.
                * apply eqb_eq in E. rewrite prop_to_prop in E.
                rewrite prop_to_prop in E. rewrite <- E in H'.
                unfold atoms_of. rewrite H'.
                rewrite Bool.orb_true_l. reflexivity.
            + unfold ∔̂ in H. rewrite E in H. exists p'; split.
                * rewrite union_comm. unfold In. unfold ∪.
                rewrite PeanoNat.Nat.add_comm. apply le_add_r2. assumption.
                * assumption.
        - destruct (PeanoNat.Nat.eqb q (prop_to_nat p')) eqn: E.
        + exists (p ⊃ q); split.
            *unfold In. rewrite add_union_extrac.
            unfold ∔̂. rewrite PeanoNat.Nat.eqb_refl.
            apply le_S_one.
            * apply eqb_eq in E. rewrite prop_to_prop in E.
            rewrite prop_to_prop in E. rewrite <- E in H'.
            unfold atoms_of. rewrite H'. rewrite Bool.orb_true_r. reflexivity.
        + unfold ∔̂ in H. rewrite E in H. exists p'; split.
            * rewrite add_union_extrac. rewrite union_comm.
            rewrite <- add_union_extrac. rewrite union_comm.
            unfold In. unfold ∪.
            rewrite PeanoNat.Nat.add_comm. apply le_add_r2. assumption.
            * assumption.
```

```
Qed.
```

□

لم ۶۵.۴ (atomin_impr).

$$\forall s_1\ a\ p\ q, atom\_in(s_1 \hat{+} q)a \to atom\_in(s \hat{+} (p \supset q), a)$$

لم ۶۶.۴ (atomin_impl).

$$\forall s_1\ a\ p\ q, atom\_in(s_1 \hat{+} p)a \to atom\_in(s \hat{+} (p \supset q), a)$$

لم ۶۷.۴ (atomsof_or_destruct).

$$\forall p\ p'\ x, atoms\_of((p \vee p'), x) \to (atoms\_of(p, x) \vee atoms\_of(p', x))$$

لم ۶۸.۴ (atomsof_and_destruct).

$$\forall p\ p'\ x, atoms\_of((p \wedge p'), x) \to (atoms\_of(p, x) \vee atoms\_of(p', x))$$

لم ۶۹.۴ (atomsof_imp_destruct).

$$\forall p\ p'\ x, atoms\_of((p \supset p'), x) \to (atoms\_of(p, x) \vee atoms\_of(p', x))$$

لم‌های زیر حالاتی خاص از لم‌های بالا هستند که در اثبات قضیه‌ی درونیابی در منطق شهودی به کار می‌آیند.

لم ۷۰.۴ (atomin_singl_add).

$$\forall s\ a\ p, atom\_in(\{\{a\}\}, p) \to atom\_in((s \hat{+} a), p)$$

لم ۷۱.۴ (option_add_emp).

$$\forall op, option\_add(\emptyset, op) = option\_to\_multiset(op)$$

لم ۷۲.۴ (atomin_option_add).

$$\forall G\ a\ op, atom\_in(G, a) \to atom\_in(G \hat{+} a, p)$$

لم ۷۳.۴ (atomin_option_addr).

$$\forall G\ a\ op, atom\_in(option\_to\_multiset(op), a) \to atom\_in(option\_add(G, op), a)$$

لم ۷۴.۴ (atomin_option_destruct).

$$\forall G\ op\ a, atom\_in(option\_add(G, op), a) \leftrightarrow$$

$$(atom\_in(G, a) \vee atom\_in(option\_to\_multiset(op), a))$$

لم ۷۵.۴ (atomin_add_option).

$$\forall G\ op\ a\ p, atom\_in(option\_add(G, op), a)$$

$$\rightarrow atom\_in(option\_add((G \hat{+} p), op), a)$$

لم ۷۶.۴ (atomin_add_destruct).

$$\forall G\ a\ p, atom\_in((G \hat{+} p), a) \leftrightarrow$$

$$atom\_in(G, a) \vee atom\_in(\{\{p\}\}, a)$$

لم ۷۷.۴ (atomin_add_double_option).

$$\forall G\ op\ a\ p, atom\_in((option\_add((G \hat{+} p) \hat{+} p), op), a) \rightarrow$$

$$atom\_in((option\_add(G \hat{+} p), op), a).$$

فصل ۵

اثبات قضیه‌ی درونیابی کریگ در منطق گزاره‌ای کلاسیک

۱.۵ دستگاه استنتاجی حساب رشته‌ها برای منطق گزاره‌ای کلاسیک

از آنجا که توضیحات کلی مرتبط با حساب رشته‌ها در فصل ۳ آورده شده‌است در این بخش تنها به معرفی دستگاه استنتاج بسنده می‌کنیم.

تذکر ۱.۵. *واضح‌ترین تفاوت دستگاه استنتاجی ما با دستگاه‌های حساب رشته‌ای مرسوم که در اکثر کتاب‌ها ارائه می‌شود. قاعده $\bot$ است. در اکثر مراجع به‌جای این قاعده، قاعده نقیض چپ و راست به شکل زیر معرفی می‌شود:*

$$\neg l\ \frac{\Gamma \dashrightarrow p, \Delta}{\Gamma, \neg p \dashrightarrow \Delta} \qquad \neg r\ \frac{\Gamma, p \dashrightarrow \Delta}{\Gamma \dashrightarrow \neg p, \Delta}$$

دلیل اینگونه معرفی ما این است که همانطور که قبلا اشاره شد در کتب ریاضی اهمیت چندانی ندارد که مثلا $\bot$ از قواعد استنتاج قابل دستیابی به راحتی باشد؛ ولی در اثبات، ما باید بعضی از درونیاب‌ها را با $\bot$ و $\top$ بسازیم پس این امر که قواعد استنتاج مستقیما و آسان این دو گزاره را به ما بدهند، برایمان مهم است. علاوه بر این، داشتن این قاعده‌ی ساده به‌جای دو قاعده‌ی دیگر خود اثبات قضیه درونیابی را کوتاه‌تر می‌کند.

تذکر ۲.۵. *قاعده $\bot$ تنها قاعده‌ای است که می‌تواند مجموعه زمینه نامتناهی خروجی دهد.*

تذکر ۳.۵. *با اینکه قاعده نقیض راست به‌راحتی قابل‌اثبات است (و در کد نیز اثبات آن موجود است) ولی برای قاعده نقیض چپ نیاز به تعریف مجموعه زمینه متناهی داریم که به دلیل پیچیدگی از اثبات آن صرف نظر شده‌است.*

$$\frac{}{A \dashrightarrow A}\ \text{Axiom}$$

$$\frac{\Gamma \dashrightarrow \Delta}{\Gamma, p \dashrightarrow \Delta}\ \text{lW} \qquad \frac{\Gamma \dashrightarrow \Delta}{\Gamma \dashrightarrow \Delta, p}\ \text{rW}$$

$$\frac{\Gamma, p, p \dashrightarrow \Delta}{\Gamma, p \dashrightarrow \Delta}\ \text{lC} \qquad \frac{\Gamma \dashrightarrow \Delta, p, p}{\Gamma \dashrightarrow \Delta, p}\ \text{rC}$$

$$\frac{}{\bot \dashrightarrow A}\ \bot$$

$$\frac{\Gamma, b \dashrightarrow \Delta}{\Gamma, a \wedge b \dashrightarrow \Delta}\ \wedge l2 \qquad \frac{\Gamma, a \dashrightarrow \Delta}{\Gamma, a \wedge b \dashrightarrow \Delta}\ \wedge l1$$

$$\frac{\Gamma \dashrightarrow \Delta, a \qquad \Gamma \dashrightarrow \Delta, b}{\Gamma \dashrightarrow \Delta, a \wedge b}\ \wedge r$$

$$\frac{\Gamma, a \dashrightarrow \Delta \qquad \Gamma, b \dashrightarrow \Delta}{\Gamma, a \vee b \dashrightarrow \Delta}\ \vee l$$

$$\frac{\Gamma \dashrightarrow \Delta, b}{\Gamma \dashrightarrow \Delta, a \vee b}\ \vee r2 \qquad \frac{\Gamma \dashrightarrow \Delta, a}{\Gamma \dashrightarrow \Delta, a \vee b}\ \vee r1$$

$$\frac{\Gamma \dashrightarrow \Delta, a \qquad \Gamma, b \dashrightarrow \Delta}{\Gamma, a \supset b \dashrightarrow \Delta}\ \supset l$$

$$\frac{\Gamma, a \dashrightarrow \Delta, b}{\Gamma \dashrightarrow \Delta, a \supset b}\ \supset r$$

شکل ۱.۵: Sequent calculus formulation for propositional classical logic

تذکر ۴.۵. *ما در کد، برای استفاده از قضیه استقرا نیاز داریم که به شکلی، ساختار درخت اثبات را ذخیره کنیم برای اینکار ما نوعی به اسم Node معرفی می‌کنیم که در اصل شکلی بسیار ساده از درخت دودویی است و در هر سازنده از دستگاه استنتاج، نحوه ساخته‌شدن آن گره از درخت اثبات از گره‌های بالایی را اضافه می‌کنیم.*

```
Inductive Node : Type :=
|leaf : Node
|onen: Node -> Node
|twon: Node -> Node -> Node.
```

سازنده leaf نماد گرهی است که هیچ پدری ندارد مثل وقتی که از قاعده Axiom یا $\bot$ استفاده می‌کنیم از نماد $\odot M$ برای نشان دادن سازنده onen یا گرهی که یک پدر دارد و پدرش M است استفاده می‌کنیم مثل قاعده $\wedge l1$ و از سازنده twon با نماد $M_1 \asymp M_2$ برای نشان دادن گرهی که دو پدر دارد و پدرش M_1 و M_2 هستند استفاده می‌کنیم مثل استفاده از قاعده $\wedge r$.

تذکر ۵.۵. *برای منطق کلاسیک از نماد --- برای نماد رشته حساب رشته‌ها استفاده می‌کنیم.*

تعریف ۶.۵. *ما در کد، حساب رشته‌ها را یک نوع استقرایی تعریف می‌کنیم که یک گره و دو مجموعه مکرر می‌گیرد و یک گزاره خروجی می‌دهد.*

```
Inductive LKS : Node -> multiset -> multiset -> Prop :=
(* Axiom : A |- A *)
(* 1 *)
|LKSA: forall p: prop, {{p}} --→ {{p}} >< leaf
(* Structral rules *)
(* Weakening *)
(* 2 *)
|LKrW: forall G D p n, G --→ D >< n -> G --→ (D ∔ p) >< (⊙ n)
(* 3 *)
|LKlW: forall G D p n, G --→ D >< n ->  (G ∔ p) --→ D >< (⊙ n)
(* Contraction *)
(* 4 *)
|LKrC: forall G D p n, G --→ (D ∔ p) ∔ p >< n  -> G --→ (D ∔ p) >< (⊙ n)
(* 5 *)
```

```
|LKlC: forall G D p n, (G ⨥ p) ⨥ p --> D >< n  -> (G ⨥ p) --> D >< (⊙ n)
(* Logical Rules *)
(* Conjunction *)
(* 6 *)
|LKrA: forall G D (a b : prop) m n,
G --> (D ⨥ a) >< n -> G --> (D ⨥ b) >< m -> G  --> (D ⨥ (a ∧ b)) >< (m ≍ n)
(* 7 *)
|LKl1A: forall G D (a b : prop) n,
(G ⨥ a) --> D >< n -> (G ⨥ (a ∧ b)) --> D >< (⊙ n)
(* 8 *)
|LKl2A: forall G D (a b : prop) n,
(G ⨥ b) --> D >< n -> (G ⨥ (a ∧ b)) --> D >< (⊙ n)
(* Disjunction *)
(* 9 *)
|LKr1O: forall G D (a b : prop) n,
G --> (D ⨥ a) >< n -> G  --> (D ⨥ (a ∨ b)) >< (⊙ n)
(* 10 *)
|LKr2O: forall G D (a b : prop) n,
G --> (D ⨥ b) >< n -> G  --> (D ⨥ (a ∨ b)) >< (⊙ n)
(* 11 *)
|LKlO: forall G D (a b : prop) m n,
(G ⨥ a) --> D >< n -> (G ⨥ b) --> D >< m -> (G ⨥ (a ∨ b)) --> D >< (m ≍ n)
(* Bot *)
(* 12 *)
| LKB : forall D, {{ ⊥ }} --> D >< leaf
(* Implication *)
(* 13 *)
|LKrI: forall G D (a b : prop) n, (G ⨥ a) --> (D ⨥ b) >< n
-> G  --> (D ⨥ (a ⊃ b)) >< (⊙ n)
(* 14 *)
|LKlI: forall G D (a b : prop) m n,
G --> (D ⨥ a) >< n -> (G ⨥ b) --> D >< m -> (G ⨥ (a ⊃ b)) --> D >< (m ≍ n)
where G --> p >< n := (LKS n G p).
```

۲.۵ اثبات قضیه‌ی درونیابی در منطق گزاره‌ای کلاسیک

برای اثبات قضیه‌ی درونیابی به روش مایهارا باید ابتدا لم زیر را ثابت کنیم:

لم ۷.۵ (لم درونیابی)**.** *برای هر مجموعه مکرر* $G1, G2, D1, D2$ *که داشته باشیم*

$$G1, G2 \dashrightarrow D1, D2$$

وجود دارد درونیابی به نام c *به طوری که*

$$G1 \dashrightarrow D1, c \tag{۱.۵}$$

$$G2, c \dashrightarrow D2 \tag{۲.۵}$$

$$atoms\_incl\ c\ (G1 \cup D1)\ (G2 \cup D2) \tag{۳.۵}$$

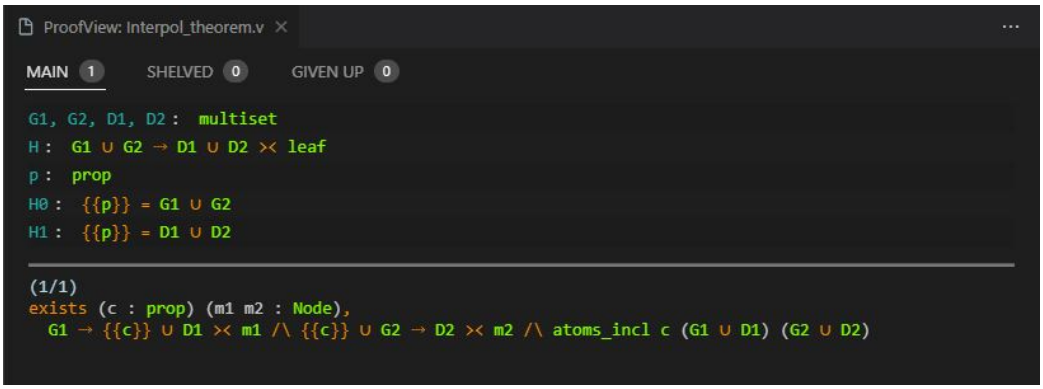
```
Theorem LK_Interpol_strong: forall n (G1 G2 D1 D2 : multiset),
G1 ∪ G2 --> D1 ∪ D2 >< n ->
(exists (c : prop) m1 m2, G1 --> {{c}} ∪ D1 >< m1 /\ {{c}} ∪ G2 --> D2 >< m2
/\ (atoms_incl c (G1 ∪ D1) (G2 ∪ D2))).
```

اثبات. این قضیه را با استقرا روی طول درخت اثبات، ثابت می‌کنیم بنابراین باید درونیاب را برای هر ۱۴ قاعده استنتاج نتیجه بگیریم. این قواعد را به ترتیبی که در کد اثبات شده‌است، اثبات می‌کنیم یعنی اول قواعدی که از ریشه درخت استنتاج هستند، بعد قواعدی که از یک پدر نتیجه شده‌اند به ترتیب تعریف شدنشان و سپس قواعدی که از دو پدر نتیجه شده‌اند.

۱. اگر درخت استنتاج با قاعده Axiom پایان یافته باشد داریم:

$$\text{Axiom}\ \frac{}{G1, G2 \dashrightarrow D1, D2}$$

در بخش proofview اثبات‌یار به طور مشابه داریم:



با اجرای لم ۴۳.۴ روی فرض های H۰ و H۱ به این نتیجه می‌رسیم که گزاره p یا در G۱ است یا در G۲، و به طور مشابه برای D۱ و D۲ پس در مجموع ۴ حالت خواهیم داشت.

(آ) اگر داشته باشیم $G1 = \{\{p\}\}$ و $G2 = \emptyset$ و $D1 = \{\{p\}\}$ و $D2 = \emptyset$ آنگاه درونیابمان $\bot$ است.
برای اثبات شرط ۱.۵ داریم:

$$G1 \dashrightarrow D1, \bot$$

که به شکل زیر اثبات می‌شود:

$$\text{rW}\ \dfrac{\text{Axiom}\ \dfrac{}{p \dashrightarrow p}}{p \dashrightarrow p, \bot}$$

برای اثبات شرط ۲.۵ داریم:

$$\bot \dashrightarrow$$

که به شکل زیر اثبات می‌شود:

$$\dfrac{\bot}{\bot \dashrightarrow}$$

شرط ۳.۵ هم به‌سادگی اثبات می‌شود زیرا تعریف ۵۷.۴ به تعریف ۵۶.۴ شکسته می‌شود و آن هم به تعریف ۵۵.۴ که برای $\bot$ به ما false می‌دهد.

(ب) اگر داشته باشیم $G1 = \{\{p\}\}$ و $G2 = \emptyset$ و $D2 = \{\{p\}\}$ و $D1 = \emptyset$ درونیابمان p است. برای اثبات شرط ۱.۵ داریم:

$$G1 \dashrightarrow D1, p$$

که به شکل زیر اثبات می‌شود:

$$\text{Axiom}\ \frac{}{p \dashrightarrow p}$$

به طور مشابه برای شرط ۲.۵ داریم:

$$G2, p \dashrightarrow D2$$

که دوباره اثبات آن به شکل زیر است:

$$\text{Axiom}\ \frac{}{p \dashrightarrow p}$$

برای شرط ۳.۵ باید ثابت کنیم برای هر اتم، گزاره‌ای مثل x در سمت چپ و راست وجود دارد که اگر $atom\_in(p, a)$ صادق باشد $atom\_in(\bar{p}, a)$ صادق است، در هر دو حالت قرار می‌دهیم $x = \bar{p}$ و این شرط اثبات می‌شود.

(ج) اگر داشته باشیم $D1 = \{\{p\}\}$ و $G1 = \emptyset$ و $G2 = \{\{p\}\}$ و $D2 = \emptyset$ درونیابمان $\neg p$ است. برای اثبات شرط ۱.۵ داریم.

$$G1 \dashrightarrow D1, \neg p$$

که به شکل زیر اثبات می‌شود:

$$\supset r\ \frac{\text{rW}\ \dfrac{\text{Axiom}\ \dfrac{}{p \dashrightarrow p}}{p \dashrightarrow p, \bot}}{\dashrightarrow p, p \supset \bot}$$

برای اثبات شرط ۲.۵ داریم

$$G2, \neg p \dashrightarrow D2$$

که به شکل زیر اثبات می‌شود:

$$\supset l\ \frac{\text{lW}\ \dfrac{\bot\ \dfrac{}{\bot \dashrightarrow}}{p, \bot \dashrightarrow} \qquad \text{Axiom}\ \dfrac{}{p \dashrightarrow p}}{p, p \supset \bot \dashrightarrow}$$

اثبات شرط ۳.۵ مشابه ۱ب است.

(د) اگر داشته باشیم $G2 = \{\{p\}\}$ و $G1 = \emptyset$ و $D2 = \{\{p\}\}$ و $D1 = \emptyset$ درونیاب مدنظر ما در این حالت $\top$ است. برای اثبات شرط ۱.۵ داریم

$$G1 \dashrightarrow D1, \top$$

که اثبات آن به شرح زیر است:

$$\supset r \dfrac{\dfrac{}{\bot \dashrightarrow \bot}\,\bot}{\dashrightarrow \bot \supset \bot}$$

برای اثبات شرط ۲.۵ داریم:

$$G2, \top \dashrightarrow D2$$

که اثبات آن به شرح زیر است:

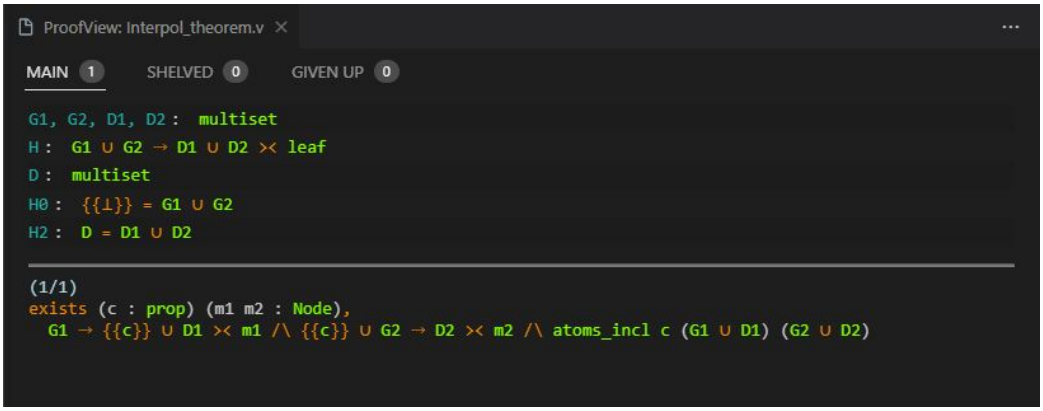
$$\mathrm{lW} \dfrac{\mathrm{Axiom}\dfrac{}{p \dashrightarrow p}}{p, \top \dashrightarrow p}$$

اثبات شرط ۳.۵ مشابه ۱آ است.

۲. اگر اثبات با قاعده $\bot$ تمام شود داریم:

$$\bot \dfrac{}{\Gamma_1 \cup \Gamma_2 \dashrightarrow \Delta_1 \cup \Delta_2}$$

و proofview اثبات‌یار تصویر زیر را به ما نشان می‌دهد:



با اجرای لم ۴۳.۴ روی فرض $H0$ به این نتیجه می‌رسیم که $\bot$ یا در G۱ است یا در G۲ پس دو حالت داریم.

(آ) اگر $G1 = \{\{\bot\}\}$ و $G2 = \emptyset$ آنگاه درونیاب مدنظر ما $\bot$ است. برای اثبات شرط ۱.۵ داریم:

$$G1 \dashrightarrow D1, \bot$$

که با یک بار اجرای قاعده استنتاج $\bot$ ثابت می‌شود. برای شرط ۲.۵ داریم:

$$G2, \bot \dashrightarrow D2$$

این هم با یک بار اجرای قاعده $\bot$ ثابت می‌شود. اثبات شرط ۳.۵ مشابه ۱آ است.

(ب) اگر $G2 = \{\{\bot\}\}$ و $G1 = \emptyset$ آنگاه درونیاب مدنظر ما $\top$ است. برای اثبات شرط ۱.۵ داریم:

$$G1 \dashrightarrow D1, \top$$

که به شکل زیر اثبات می‌شود.

$$\supset r \dfrac{\bot \dfrac{}{\bot \dashrightarrow D1, \bot}}{\dashrightarrow D1, \bot \supset \bot}$$

برای شرط ۲.۵ داریم:

$$G2, \top \dashrightarrow D2$$

که به شکل زیر اثبات می‌شود:

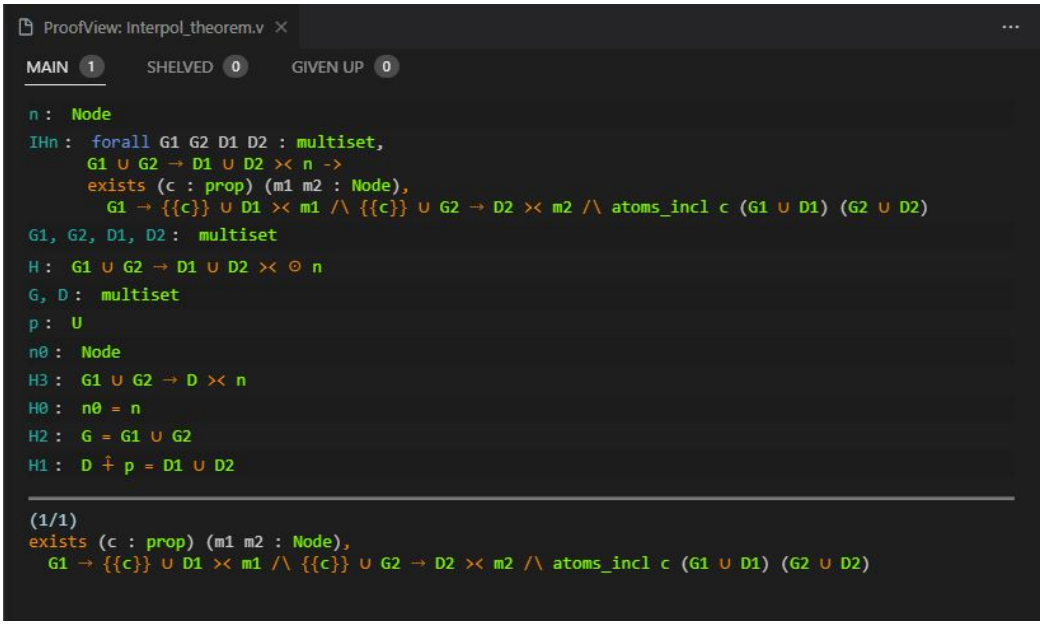
$$\mathrm{lW} \dfrac{\bot \dfrac{}{\bot \dashrightarrow D2}}{\bot, \bot \supset \bot \dashrightarrow D2}$$

اثبات شرط ۳.۵ مشابه ۱آ است.

۳. اگر درخت اثبات به قاعده rW ختم شود داریم:

$$\mathrm{rW} \dfrac{G1, G2 \dashrightarrow D1', D2'}{G1, G2 \dashrightarrow D1', D2', p}$$

و proofview اثبات‌یار فرضیات را به شکل زیر به ما نشان می‌دهد:



با یک بار اجرای لم ۴۸.۴ روی فرض $H1$ و اجرای لم ۴۹.۴ روی نتایج آن مسئله به دو حالت تقسیم می‌شود.

(آ) اگر $p \in D1$ باشد آنگاه داریم $\exists D', D' \hat{+} p = D1$ و بنا به لم ۵۱.۴ و لم ۵۰.۴ خواهیم داشت $D = D' \cup D2$ که می‌توانیم از $D', D2$ در فرض استقرا (IHn) استفاده کنیم. از فرض استقرا خواهیم داشت:

$$\exists c, G1 \dashrightarrow D', c$$

$$G2, c \dashrightarrow D2$$

$$atom\_incl\ c\ (G1 \cup D')\ (G2 \cup D2)$$

c حاصل از فرض استقرا را درونیاب در نظر می‌گیریم. برای اثبات ۱.۵ داریم:

$$\text{rW}\ \frac{\dfrac{G1 \dashrightarrow D', c}{G1 \dashrightarrow D', c, p}}{G1 \dashrightarrow D1, c}$$

شرط ۲.۵ جزو فرض استقرا است. برای اثبات شرط ۳.۵ باید ثابت کنیم برای هر اتمی مثل a که داشته باشیم $atoms\_of(c, a)$ آنگاه داریم:

$$atom\_in((G1 \cup (D' \hat{+} p)), a)$$

$$atom\_in((G2 \cup D2), a)$$

فرض استقرا به ما می‌دهد که برای هر اتم که $atoms\_of(c, a)$ داریم.

$$atom\_in((G1 \cup D'), a)$$

$$atom\_in((G2 \cup D2), a)$$

پس شرط دوم این شرط از فرض استقرا نتیجه می‌شود برای شرط اول هم از لم ۵۸.۴ استفاده می‌کنیم.

(ب) اگر $p \in D2$ باشد آنگاه داریم $\exists D', D' \hat{+} p = D2$ و بنا به لم ۵۱.۴ و لم ۵۰.۴ خواهیم داشت $D = D' \cup D1$ که می‌توانیم از $D1, D'$ در فرض استقرا (IHn) استفاده کنیم. از فرض استقرا خواهیم داشت:

$$\exists c, G1 \dashrightarrow D1, c$$

$$G2, c \dashrightarrow D'$$

$$atom\_incl\ c\ (G1 \cup D1)\ (G2 \cup D')$$

گزاره c از فرض استقرا را درونیاب در نظر می‌گیریم. شرط ۱.۵ جزو فرض استقرا است برای شرط ۲.۵ از فرض استقرا خواهیم داشت:

$$\text{rW}\ \frac{\dfrac{G2, c \dashrightarrow D'}{G2, c \dashrightarrow D', p}}{G2, c \dashrightarrow D2}$$

برای اثبات شرط ۳.۵ باید ثابت کنیم برای هر اتمی مثل a که داشته باشیم $atoms\_of(c, a)$ آنگاه داریم:

$$atom\_in((G1 \cup D1), a)$$

$$atom\_in((G2 \cup (D' \hat{+} p)), a)$$

فرض استقرا به ما می‌دهد که برای هر اتم که $atoms\_of(c, a)$ داریم.

$$atom\_in((G1 \cup D1), a)$$

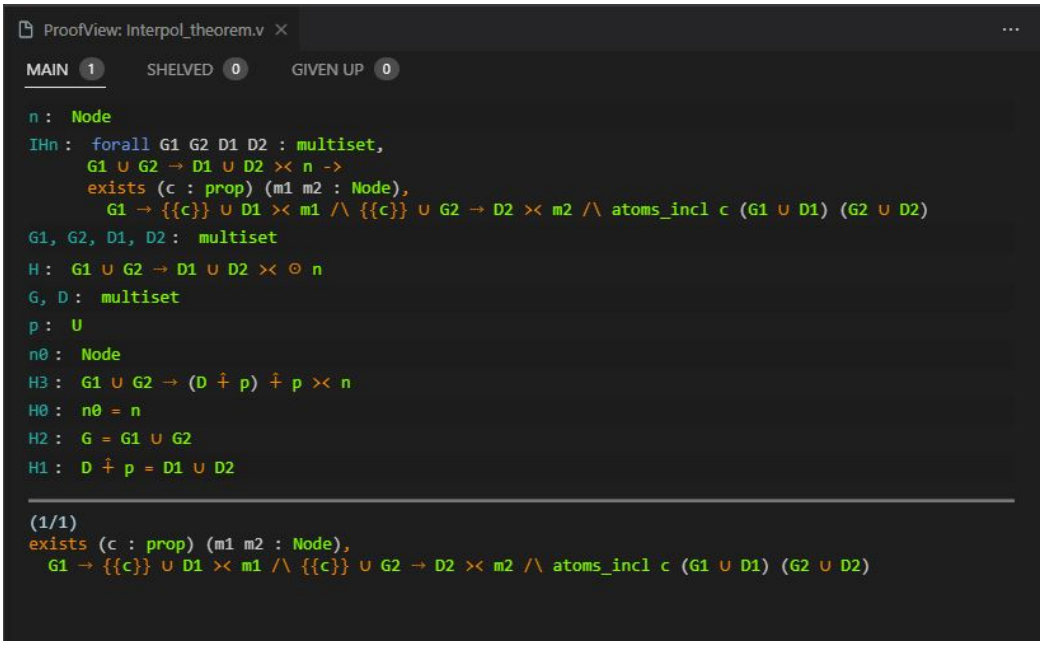
$$atom\_in((G2 \cup D'), a)$$

مشابه قبل با لم ۵۸.۴ این شرط اثبات می‌شود.

۴. اگر درخت اثبات به قاعده rW ختم شود کاملا مشابه ۳ اثبات می‌شود.

۵. اگر درخت اثبات به قاعده rC ختم شود خواهیم داشت:

$$\text{rC}\ \frac{G1, G2 \dashrightarrow D1', D2', p, p}{G1, G2 \dashrightarrow D1', D2', p}$$



مشابه قبل با لم ۴۸.۴ و ۴۹.۴ مسئله را به دو حالت تقسیم می‌کنیم.

(آ) اگر $p \in D1$ باشد آنگاه داریم $\exists D', D' \hat{+} p = D1$ و بنا به لم ۵۱.۴ و لم ۵۰.۴ خواهیم داشت $D = D' \cup D2$ که می‌توانیم از $D', D2$ در فرض استقرا (IHn) استفاده کنیم. از فرض استقرا خواهیم داشت:

$$\exists c, G1 \dashrightarrow D', p, p, c$$

$$G2, c \dashrightarrow D2$$

$$atom\_incl\ c\ (G1 \cup (D' \hat{+} p \hat{+} p))\ (G2 \cup D2)$$

شرط ۱.۵ را به صورت زیر اثبات می‌کنیم:

$$\text{rC}\ \frac{\dfrac{G1 \dashrightarrow D', p, p, c}{G1 \dashrightarrow D', p, c}}{G1 \dashrightarrow D1, c}$$

درونیاب را c در نظر می‌گیریم شرط ۲.۵ جزو فرض استقرا است و شرط ۳.۵ با فرض استقرا و لم ۵۹.۴ اثبات می‌شود.

(ب) اگر $p \in D2$ باشد آنگاه داریم $\exists D', D' \hat{+} p = D2$ و بنا به لم ۵۱.۴ و لم ۵۰.۴ خواهیم داشت $D = D' \cup D1$ که می‌توانیم از $D1, D'$ در فرض استقرا (IHn) استفاده کنیم. از فرض استقرا خواهیم داشت:

$$\exists c, G1 \dashrightarrow D1, c$$

$$G2, c \dashrightarrow D', p, p$$

$$atom\_incl\ c\ (G1 \cup D1)\ (G2 \cup (D' \hat{+} p \hat{+} p))$$

درونیاب را c در نظر می‌گیریم شرط ۱.۵ جزو فرض استقرا است و شرط ۲.۵ به شکل زیر اثبات می‌شود:

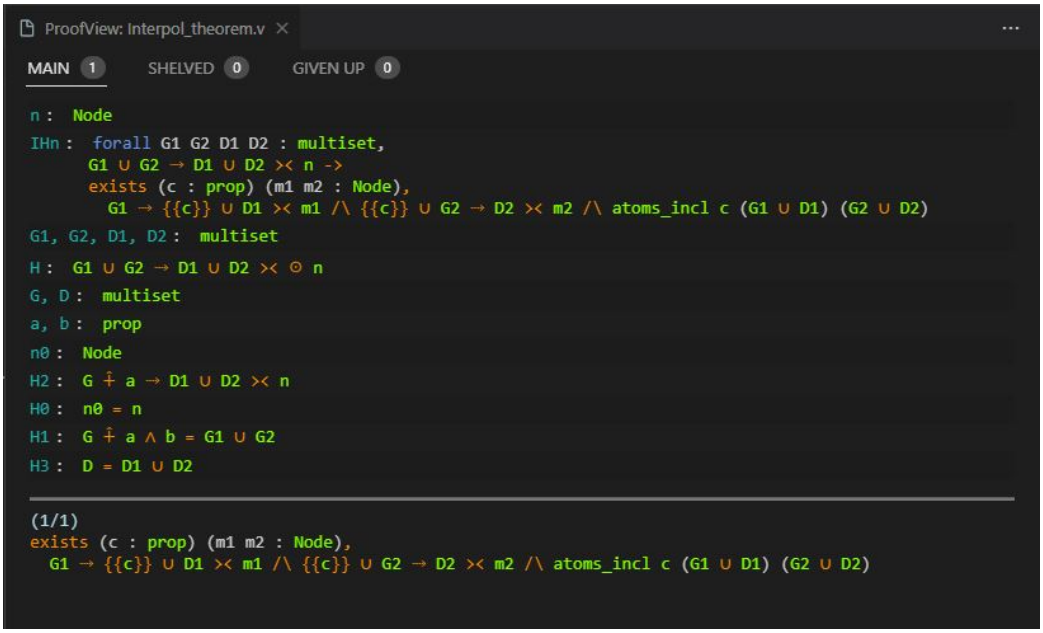
$$\text{rC}\ \frac{\dfrac{G2, c \dashrightarrow D', p, p}{G2, c \dashrightarrow D', p}}{G2, c \dashrightarrow D2}$$

شرط ۳.۵ مشابه بخش قبل با لم ۵۹.۴ و فرض استقرا اثبات می‌شود.

۶. اگر درخت اثبات به قاعده lC ختم شود اثبات دقیقا مشابه ۵ است.

۷. اگر درخت اثبات به قاعده $\wedge l1$ ختم شود داریم:

$$\wedge l1\ \frac{G1', G2', a \dashrightarrow D1, D2}{G1, G2, a \wedge b \dashrightarrow D1, D2}$$



با اجرای لم ۴۸.۴ روی فرض $H0$ به این نتیجه می‌رسیم که $a \wedge b$ یا در G۱ است یا در G۲ پس دو حالت داریم.

(آ) اگر $a \wedge b \in G1$ باشد آنگاه داریم $\exists G', G' \hat{+} a \wedge b = G1$ و بنا به لم ۵۱.۴ و لم ۵۰.۴ خواهیم داشت $G = G' \cup G2$ که می‌توانیم از $G', G2$ در فرض استقرا (IHn) استفاده کنیم. از فرض استقرا خواهیم داشت:

$$\exists c, G', a \dashrightarrow D1, c$$

$$G2, c \dashrightarrow D2$$

$$atom\_incl\ c\ (G' \hat{+} a \cup D1)\ (G2 \cup D2)$$

درونیاب را c قرار می‌دهیم برای شرط ۱.۵ داریم:

$$\wedge l1\ \frac{\dfrac{G', a \dashrightarrow D1, c}{G', a \wedge b \dashrightarrow D1, c}}{G1 \dashrightarrow D1, c}$$

شرط ۲.۵ مستقیما از فرض استقرا نتیجه می‌شود. برای شرط ۳.۵ از فرض استقرا داریم که برای هر a اگر $atoms\_of(c, a)$ آنگاه:

$$atom\_in((G' \hat{+} a) \cup D1)a$$

$$atom\_in(G2 \cup D2)a$$

و باید اثبات کنیم برای هر a اگر $atoms\_of(c, a)$ داریم:

$$atom\_in((G' \hat{+} (a \wedge b)) \cup D1)a$$

$$atom\_in(G2 \cup D2)a$$

با استفاده از لم ۶۰.۴ از فرض استقرا این حکم نتیجه می‌شود.

(ب) اگر $a \wedge b \in G2$ باشد آنگاه داریم $\exists G', G' \hat{+} a \wedge b = G2$ و بنا به لم ۵۱.۴ و لم ۵۰.۴ خواهیم داشت $G = G' \cup G1$ که می‌توانیم از $G', G1$ در فرض استقرا (IHn) استفاده کنیم. از فرض استقرا خواهیم داشت:

$$\exists c, G1 \dashrightarrow D1, c$$

$$G', a, c \dashrightarrow D2$$

$$atom\_incl\ c\ (G1 \cup D1)\ (G' \hat{+} a \cup D2)$$

درونیاب را c قرار می‌دهیم شرط ۱.۵ از فرض استقرا نتیجه می‌شود برای شرط ۲.۵ داریم:

$$\wedge l1\ \frac{\dfrac{G', a, c \dashrightarrow D2}{G', a \wedge b, c \dashrightarrow D2}}{G2, c \dashrightarrow D2}$$

برای شرط ۳.۵ از فرض استقرا داریم که برای هر a اگر $atoms\_of(c, a)$ آنگاه:

$$atom\_in(G1 \cup D1)a$$

$$atom\_in(G' \hat{+} a \cup D2)a$$

و باید اثبات کنیم برای هر a اگر $atoms\_of(c, a)$ داریم:

$$atom\_in(G1 \cup D1)a$$

$$atom\_in(G' \hat{+} (a \wedge b) \cup D2)a$$

با استفاده از لم ۶۰.۴ از فرض استقرا این حکم نتیجه می‌شود.

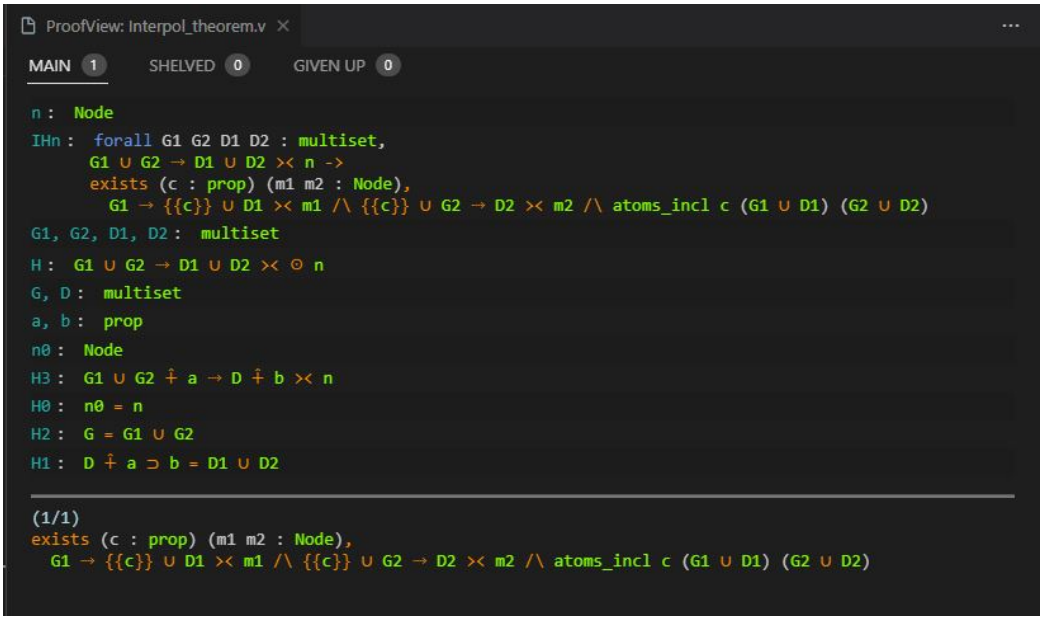
۸. اگر درخت اثبات به قاعده $\wedge l2$ ختم شود اثبات کاملا مشابه ۷ است با این تفاوت که برای شرط ۳.۵ باید از لم ۶۱.۴ استفاده کنیم.

۹. اگر درخت اثبات به قاعده $\vee r1$ ختم شود اثبات کاملا مشابه ۷ است با این تفاوت که برای شرط ۳.۵ باید از لم ۶۲.۴ استفاده کنیم.

۱۰. اگر درخت اثبات به قاعده $\vee r2$ ختم شود اثبات کاملا مشابه ۷ است با این تفاوت که برای شرط ۳.۵ باید از لم ۶۳.۴ استفاده کنیم.

۱۱. اگر درخت اثبات به قاعده $r \supset$ ختم شود داریم:

$$\supset r \frac{G1, G', a \dashrightarrow b, D1', D2'}{G1, G2 \dashrightarrow D1, D2, a \supset b}$$



مشابه قبل این حالت به دو حالت تقسیم می‌شود:

(آ) اگر داشته باشیم $a \supset b \in D1$ آنگاه خواهیم داشت $\exists D', D' \hat{+}(a \supset b) = D1$ با قرار دادن $(G1 \hat{+} a), G2, (D' \hat{+} b), D2$ در فرض استقرا نتیجه زیر به دست می‌آید:

$$\exists c, (G1 \hat{+} a) \dashrightarrow (D' \hat{+} b), c$$

$$G2, c \dashrightarrow D2$$

$$atom\_incl\ c\ ((G1 \hat{+} a) \cup (D' \hat{+} b))\ (G2 \cup D2)$$

اگر درونیاب را c قرار دهیم شرط ۱.۵ به شکل زیر اثبات می‌شود:

$$\supset r\ \frac{\dfrac{G1, a \dashrightarrow D', b, c}{G1 \dashrightarrow D', a \supset b, c}}{G1 \dashrightarrow D1, c}$$

شرط ۲.۵ از فرض استقرا نتیجه می‌شود و شرط ۳.۵ مشابه قبل اثبات می‌شود با این تفاوت که از لم ۶۴.۴ استفاده می‌کنیم.

(ب) اگر داشته باشیم $\exists D', D'\hat{+}(a \supset b) = D2$ آنگاه خواهیم داشت $a \supset b \in D2$ با قرار دادن $G1, (G2\hat{+}a), D1, (D'\hat{+}b)$ در فرض استقرا نتیجه زیر به دست می‌آید:

$$\exists c, G1 \dashrightarrow D1, c$$

$$(G2\hat{+}a), c \dashrightarrow (D'\hat{+}b)$$

$$atom\_incl\ c\ (G1 \cup D1)\ ((G2\hat{+}a) \cup (D'\hat{+}b))$$

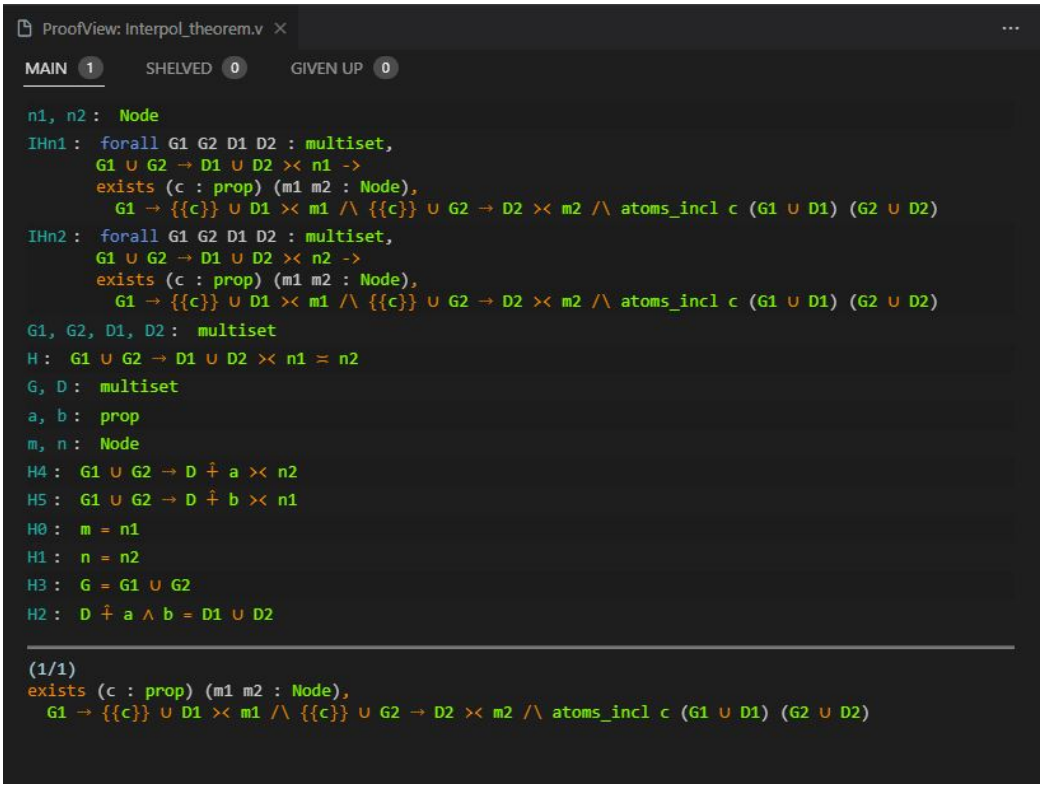
اگر درونیاب را c قرار دهیم شرط ۱.۵ از فرض استقرا نتیجه می‌شود برای شرط ۲.۵ داریم:

$$\supset r\ \frac{\dfrac{G2, a, c \dashrightarrow D', b}{G2, c \dashrightarrow D', a \supset b}}{G2, c \dashrightarrow D2}$$

و برای اثبات شرط ۳.۵ دوباره از لم ۶۴.۴ استفاده می‌کنیم.

۱۲. اگر درخت اثبات به قاعده $\wedge r$ ختم شده باشد به این معنی است که داریم:

$$\wedge r\ \frac{G1, G2 \dashrightarrow D1', D2', a \qquad G1, G2 \dashrightarrow D1', D2', b}{G1, G2 \dashrightarrow D1', D2', a \wedge b}$$



مشابه حالات قبل دو حالت داریم:

(آ) اگر $a \wedge b \in D1$ آنگاه داریم $\exists D', D' \hat{+} (a \wedge b) = D1$ نکته مهم در این حالت این است که دو فرض استقرا داریم یکی برای گزاره بالا سمت چپ و یکی برای گزاره بالا سمت راست، در فرض سمت چپ قرار می‌دهیم $G1, G2, D' \hat{+} a, D2$ و به نتایج زیر می‌رسیم:

$$\exists c, G1 \dashrightarrow D', a, c$$

$$G2, c \dashrightarrow D2$$

$$atoms\_incl\ c\ (G1 \cup (D' \hat{+} a))\ (G2 \cup D2)$$

و در فرض سمت راست قرار می‌دهیم $G1, G2, D' \hat{+} b, D2$ و نتیجه می‌گیریم:

$$\exists c', G1 \dashrightarrow D', b, c'$$

$$G2, c' \dashrightarrow D2$$

$$atoms\_incl\ c'\ (G1 \cup (D' \hat{+} b))\ (G2 \cup D2)$$

درونیاب را $c \vee c'$ در نظر می‌گیریم. برای شرط ۱.۵ اثبات به شکل زیر است:

$$\wedge r \dfrac{\vee r1 \dfrac{G1 \dashrightarrow D', a, c}{G1 \dashrightarrow D', a, c \vee c'} \quad \vee r2 \dfrac{G1 \dashrightarrow D', b, c'}{G1 \dashrightarrow D', b, c \vee c'}}{\dfrac{G1 \dashrightarrow D', a \wedge b, c \vee c'}{G1 \dashrightarrow D1, c \vee c'}}$$

برای شرط ۲.۵ اثبات به شرح زیر است.

$$\vee l \dfrac{G2, c' \dashrightarrow D2 \qquad G2, c \dashrightarrow D2}{G2, c \vee c' \dashrightarrow D2}$$

برای شرط ۳.۵ از فرض استقرای اول داریم:

$$\forall x, atoms\_of(c, x) \rightarrow (atoms\_in((G1 \cup (D' \hat{+} a)), x)$$
$$\wedge\ atoms\_in((G2 \cup D2), x))$$

و از فرض استقرای دوم داریم:

$$\forall x, atoms\_of(c', x) \rightarrow (atoms\_in((G1 \cup (D' \hat{+} b)), x)$$
$$\wedge\ atoms\_in((G2 \cup D2), x))$$

و باید ثابت کنیم برای هر x که داشته باشیم $atoms\_of(c \vee c', x)$ داریم:

$$atoms\_in((G1 \cup (D' \hat{+} a \wedge b)), x)$$
$$atoms\_in((G2 \cup D2), x)$$

با استفاده از لم ۶۷.۴ داریم که یا $atoms\_of(c, x)$ درست است یا $atoms\_of(c', x)$ در هر دو حالت حکم دوم در فرض استقرا وجود دارد اگر $atoms\_of(c, x)$ حکم اول از فرض اول استقرای اول و لم ۶۱.۴ به دست می‌آید و اگر $atoms\_of(c', x)$ حکم اول از فرض اول استقرای دوم و لم ۶۰.۴ به دست می‌آید.

(ب) اگر $a \wedge b \in D2$ آنگاه داریم $\exists D', D' \hat{+} (a \wedge b) = D2$ دوباره دو فرض استقرا داریم یکی برای گزاره بالا سمت چپ و یکی برای گزاره بالا سمت راست، در فرض سمت چپ قرار می‌دهیم $G1, G2, D1, D' \hat{+} a$ و به نتایج زیر می‌رسیم:

$$\exists c, G1 \dashrightarrow D1, c$$
$$G2, c \dashrightarrow D', a$$
$$atoms\_incl\ c\ (G1 \cup D1)\ (G2 \cup (D' \hat{+} a))$$

و در فرض سمت راست قرار می‌دهیم $G1, G2, D1, D' \hat{+} b$ و نتیجه می‌گیریم:

$$\exists c', G1 \dashrightarrow D1, c'$$

$$G2, c' \dashrightarrow D', b$$

$$atoms\_incl\ c'\ (G1 \cup D1)\ (G2 \cup (D' \hat{+} b))$$

درونیاب را $c \wedge c'$ قرار می‌دهیم. برای شرط ۱.۵ داریم:

$$\wedge r\ \frac{G1 \dashrightarrow D1, c \qquad G1 \dashrightarrow D1, c'}{G1 \dashrightarrow D1, c \wedge c'}$$

برای شرط ۲.۵ داریم:

$$\wedge r\ \frac{\wedge l1\ \dfrac{G2, c' \dashrightarrow D', a}{G2, c \wedge c' \dashrightarrow D', a} \qquad \wedge l2\ \dfrac{G2, c' \dashrightarrow D', b}{G2, c \wedge c' \dashrightarrow D', b}}{\dfrac{G2, c \wedge c' \dashrightarrow D', a \wedge b}{G2, c \wedge c' \dashrightarrow D2}}$$

برای شرط ۳.۵ از فرض استقرای اول داریم:

$$\forall x, atoms\_of(c, x) \rightarrow (atoms\_in((G1 \cup D1), x)$$

$$\wedge\ atoms\_in((G2 \cup (D' \hat{+} a)), x))$$

و از فرض استقرای دوم داریم:

$$\forall x, atoms\_of(c', x) \rightarrow (atoms\_in((G1 \cup D1), x)$$

$$\wedge\ atoms\_in((G2 \cup (D' \hat{+} b)), x))$$

و باید ثابت کنیم برای هر x که داشته باشیم $atoms\_of(c \wedge c', x)$ داریم:

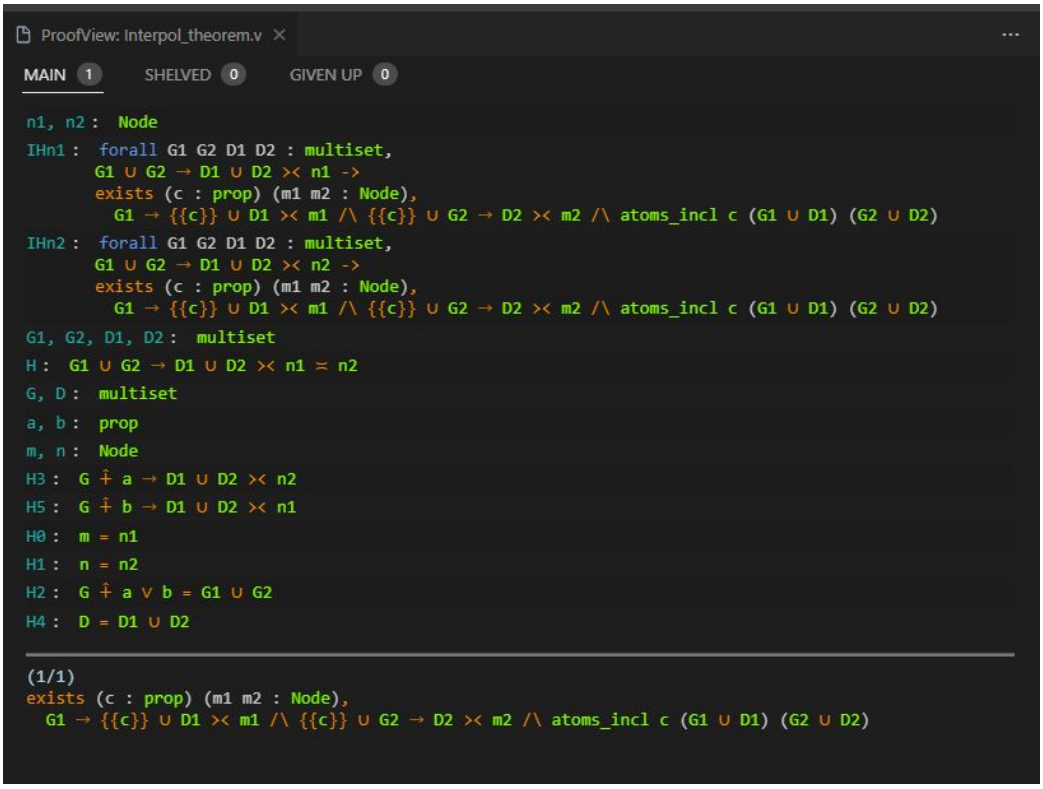
$$atoms\_in((G1 \cup D1), x)$$

$$atoms\_in((G2 \cup (D' \hat{+} a \wedge b)), x)$$

با استفاده از لم ۶۸.۴ داریم که یا $atoms\_of(c, x)$ درست است یا $atoms\_of(c', x)$ در هر دو حالت حکم اول در فرض استقرا وجود دارد اگر $atoms\_of(c, x)$ حکم دوم از فرض دوم استقرای اول و لم ۶۱.۴ به دست می‌آید و اگر $atoms\_of(c', x)$ حکم دوم از فرض دوم استقرای دوم و لم ۶۰.۴ به دست می‌آید.

۱۳. اگر درخت اثبات به قاعده $\vee l$ ختم شده باشد داریم:

$$\vee l\ \frac{G1', G2', b \dashrightarrow D1, D2 \qquad G1', G2', a \dashrightarrow D1, D2}{G1', G2', a \vee b \dashrightarrow D1, D2}$$



دو حالت ما به شرح زیر است:

(آ) اگر $a \vee b \in G1$ آنگاه مشابه قبل داریم $\exists G', G' \hat{+} (a \vee b) = G1$ در فرض استقرای اول $(G' \hat{+} b), G2, D1, D2$ را قرار می‌دهیم و خواهیم داشت:

$$\exists c,\ G', b \dashrightarrow D1, c$$

$$G2, c \dashrightarrow D2$$

$$atom\_incl\ c\ ((G' \hat{+} b) \cup D1)\ (G2 \cup D2)$$

در فرض استقرای دوم $(G' \hat{+} a), G2, D1, D2$ را قرار می‌دهیم و خواهیم داشت:

$$\exists c', G', a \dashrightarrow D1, c'$$

$$G2, c' \dashrightarrow D2$$

$$atom\_incl\ c'\ ((G' \hat{+} a) \cup D1)\ (G2 \cup D2)$$

درونیاب را $c \vee c'$ در نظر می‌گیریم. برای شرط ۱.۵ داریم:

$$\dfrac{\dfrac{\dfrac{G', a \dashrightarrow D1, c'}{G', a \dashrightarrow D1, c \vee c'}\ \vee r2 \quad \dfrac{G', b \dashrightarrow D1, c}{G', b \dashrightarrow D1, c \vee c'}\ \vee r1}{G', a \vee b \dashrightarrow D1, c \vee c'}\ \vee l}{G1 \dashrightarrow D1, c \vee c'}$$

برای شرط ۲.۵ داریم:

$$\dfrac{G2, c' \dashrightarrow D2 \qquad G2, c \dashrightarrow D2}{G2, c \vee c' \dashrightarrow D1}\ \vee l$$

برای شرط ۳.۵ از فرض استقرای اول داریم:

$$\forall x, atoms\_of(c, x) \rightarrow (atoms\_in(((G' \hat{+} a) \cup D1), x)$$

$$\wedge\ atoms\_in((G2 \cup D2), x))$$

و از فرض استقرای دوم داریم:

$$\forall x, atoms\_of(c', x) \rightarrow (atoms\_in(((G' \hat{+} b) \cup D1), x)$$

$$\wedge\ atoms\_in((G2 \cup D2), x))$$

و باید ثابت کنیم برای هر x که داشته باشیم $atoms\_of(c \vee c', x)$ داریم:

$$atoms\_in(((G' \hat{+} a \vee b) \cup D1), x)$$

$$atoms\_in((G2 \cup D2), x)$$

با استفاده از لم ۶۷.۴ داریم که یا $atoms\_of(c, x)$ درست است یا $atoms\_of(c', x)$ در هر دو حالت حکم دوم در فرض استقرا وجود دارد اگر $atoms\_of(c, x)$ حکم اول از فرض اول استقرای اول و لم ۶۳.۴ به دست می‌آید و اگر $atoms\_of(c', x)$ حکم اول از فرض اول استقرای دوم و لم ۶۲.۴ به دست می‌آید.

(ب) اگر $a \vee b \in G2$ آنگاه مشابه قبل داریم $\exists G', G'\hat{+}(a \vee b) = G2$ در فرض استقرای اول $G1, (G'\hat{+}b), D1, D2$ را قرار می‌دهیم و خواهیم داشت:

$$\exists c,\ G1 \dashrightarrow D1, c$$

$$G', b, c \dashrightarrow D2$$

$$atom\_incl\ c\ (G1 \cup D1)\ ((G'\hat{+}b) \cup D2)$$

در فرض استقرای دوم $G1, (G'\hat{+}a), D1, D2$ را قرار می‌دهیم و خواهیم داشت:

$$\exists c', G1 \dashrightarrow D1, c'$$

$$G', a, c' \dashrightarrow D2$$

$$atom\_incl\ c'\ (G1 \cup D1)\ ((G'\hat{+}a) \cup D2)$$

درونیاب را $c \wedge c'$ در نظر می‌گیریم. برای شرط ۱.۵ داریم:

$$\wedge r\ \frac{G1 \dashrightarrow D1, c' \qquad G1 \dashrightarrow D1, c}{G1 \dashrightarrow D1, c \wedge c'}$$

برای شرط ۲.۵ داریم:

$$\vee l\ \frac{\wedge l2\ \dfrac{G', a, c' \dashrightarrow D2}{G', a, c \wedge c' \dashrightarrow D2} \qquad \wedge l1\ \dfrac{G', b, c \dashrightarrow D2}{G', b, c \wedge c' \dashrightarrow D2}}{\dfrac{G', a \vee b, c \wedge c' \dashrightarrow D2}{G2, c \wedge c' \dashrightarrow D2}}$$

برای شرط ۳.۵ از فرض استقرای اول داریم:

$$\forall x, atoms\_of(c, x) \rightarrow (atoms\_in(((G'\hat{+}a) \cup D1), x)$$

$$\wedge\ atoms\_in((G2 \cup D2), x))$$

و از فرض استقرای دوم داریم:

$$\forall x, atoms\_of(c', x) \rightarrow (atoms\_in(((G'\hat{+}b) \cup D1), x)$$

$$\wedge\ atoms\_in((G2 \cup D2), x))$$

و باید ثابت کنیم برای هر x که داشته باشیم $atoms\_of(c \wedge c', x)$ داریم:

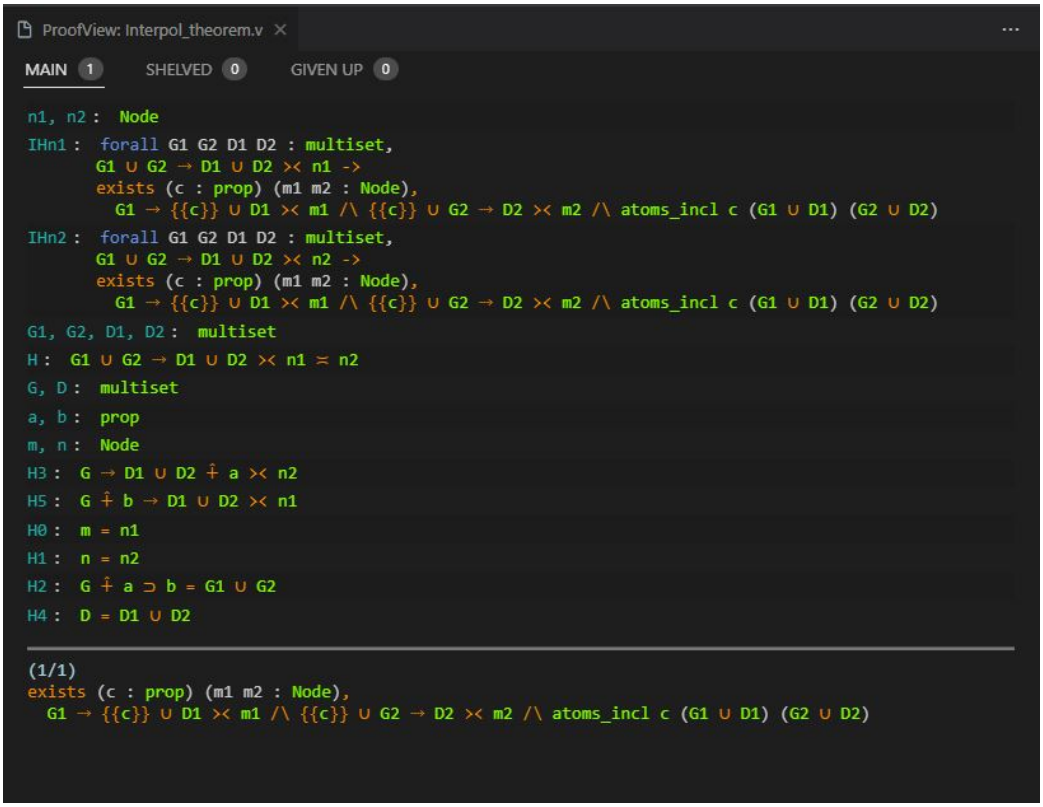
$$atoms\_in(((G'\hat{+}a \vee b) \cup D1), x)$$

$$atoms\_in((G2 \cup D2), x)$$

با استفاده از لم ۶۸.۴ داریم که یا $atoms\_of(c, x)$ درست است یا $atoms\_of(c', x)$ در هر دو حالت حکم دوم در فرض استقرا وجود دارد اگر $atoms\_of(c, x)$ حکم اول از فرض اول استقرای اول و لم ۶۳.۴ به دست می‌آید و اگر $atoms\_of(c', x)$ حکم اول از فرض اول استقرای دوم و لم ۶۲.۴ به دست می‌آید.

۱۴. اگر درخت استنتاج به قاعده $\supset l$ ختم شود داریم:

$$\supset l \frac{G1', G2' \dashrightarrow D1, D2, b \qquad G1', G2', a \dashrightarrow D1, D2}{G1', G2', a \supset b \dashrightarrow D1, D2}$$



مسئله را به دو حالت تقسیم می‌کنیم:

(آ) اگر $a \supset b \in G1$ باشد آنگاه $\exists G', G' \hat{+}(a \supset b) = G1$ در فرض استقرای اول $(G' \hat{+} a), G2, D1, D2$ را قرار می‌دهیم و فرضیات زیر را به دست می‌آوریم:

$$\exists c,\ G', a \dashrightarrow D1, c$$

$$G2, c \dashrightarrow D2$$

$$atom\_incl\ c\ ((G' \hat{+} a) \cup D1)\ (G2 \cup D2)$$

در فرض استقرای دوم $G', G2, (D1 \hat{+} b), D2$ را قرار می‌دهیم و به نتایج زیر می‌رسیم:

$$\exists c', G' \dashrightarrow D1, b, c'$$

$$G2, c' \dashrightarrow D2$$

$$atom\_incl\ c'\ (G' \cup (D1 \hat{+} b))\ (G2 \cup D2)$$

درونیاب جدید را $c \vee c'$ قرار می‌دهیم برای اثبات شرط ۱.۵ به روش زیر عمل می‌کنیم:

$$\dfrac{\dfrac{\dfrac{G' \dashrightarrow D1, b, c}{G' \dashrightarrow D1, b, c \vee c'}\ \vee r2 \qquad \dfrac{G', a \dashrightarrow D1, c}{G', a \dashrightarrow D1, c \vee c'}\ \vee r1}{G', a \supset b \dashrightarrow D1, c \vee c'}\ \supset l}{G1 \dashrightarrow D1, c \vee c'}$$

برای شرط ۲.۵ داریم:

$$\dfrac{G2, c' \dashrightarrow D2 \qquad G2, c \dashrightarrow D2}{G2, c \vee c' \dashrightarrow D2}\ \vee l$$

برای شرط ۳.۵ از فرض استقرای اول داریم:

$$\forall x, atoms\_of(c, x) \rightarrow (atoms\_in(((G' \hat{+} a) \cup D1), x)$$

$$\wedge\ atoms\_in((G2 \cup D2), x))$$

و از فرض استقرای دوم داریم:

$$\forall x, atoms\_of(c', x) \rightarrow (atoms\_in((G' \cup (D1 \hat{+} b)), x)$$

$$\wedge\ atoms\_in((G2 \cup D2), x))$$

و باید ثابت کنیم برای هر x که داشته باشیم $atoms\_of(c \vee c', x)$ داریم:

$$atoms\_in(((G' \hat{+} a \supset b) \cup D1), x)$$

$$atoms\_in((G2 \cup D2), x)$$

مشابه حالات قبل از لم ۶۷.۴ استفاده می‌کنیم که نشان می‌دهد یا $atoms\_of(c, x)$ درست است یا $atoms\_of(c', x)$ در هر دو حالت حکم دوم در فرض استقرا وجود دارد اگر $atoms\_of(c, x)$ حکم اول از فرض اول استقرای اول و لم ۶۵.۴ به دست می‌آید و اگر $atoms\_of(c', x)$ حکم اول از فرض اول استقرای دوم و لم ۶۶.۴ به دست می‌آید.

(ب) اگر $a \supset b \in G2$ باشد آنگاه $\exists G', G'\hat{+}(a \supset b) = G2$ در فرض استقرای اول $G1, (G'\hat{+}a), D1, D2$ را قرار می‌دهیم و فرضیات زیر را به دست می‌آوریم:

$$\exists c,\ G1 \dashrightarrow D1, c$$

$$G', a, c \dashrightarrow D2$$

$$atom\_incl\ c\ (G1 \cup D1)\ ((G'\hat{+}a) \cup D2)$$

در فرض استقرای دوم $G', G2, D1, (D2\hat{+}b)$ را قرار می‌دهیم و به نتایج زیر می‌رسیم:

$$\exists c', G1 \dashrightarrow D1, c'$$

$$G', c' \dashrightarrow D2, b$$

$$atom\_incl\ c'\ (G' \cup D1)\ (G2 \cup (D2\hat{+}b))$$

درونیاب جدید را $c \wedge c'$ قرار می‌دهیم برای اثبات شرط ۱.۵ به روش زیر عمل می‌کنیم:

$$\wedge r\ \frac{G1 \dashrightarrow D1, c' \qquad G1 \dashrightarrow D1, c}{G1 \dashrightarrow D1, c \wedge c'}$$

برای شرط ۲.۵ داریم:

$$\supset l\ \frac{\dfrac{\wedge l2\ \dfrac{G', c' \dashrightarrow D2, b}{G', c \wedge c' \dashrightarrow D2, b} \qquad \wedge l1\ \dfrac{G', a, c \dashrightarrow D2}{G', a, c \wedge c' \dashrightarrow D2}}{G', a \supset b, c \wedge c' \dashrightarrow D2}}{G2, c \wedge c' \dashrightarrow D2}$$

برای شرط ۳.۵ از فرض استقرای اول داریم:

$$\forall x, atoms\_of(c, x) \rightarrow (atoms\_in((G1 \cup D1), x)$$

$$\wedge\ atoms\_in(((G'\hat{+}a) \cup D2), x))$$

و از فرض استقرای دوم داریم:

$$\forall x, atoms\_of(c', x) \rightarrow (atoms\_in((G1 \cup D1), x)$$

$$\wedge\ atoms\_in((G2 \cup (D2 \hat{+} b)), x))$$

و باید ثابت کنیم برای هر x که داشته باشیم $atoms\_of(c \wedge c', x)$ داریم:

$$atoms\_in((G1 \cup D1), x)$$

$$atoms\_in(((G' \hat{+} a \supset b) \cup D2), x)$$

مشابه حالات قبل از لم ۶۸.۴ استفاده می‌کنیم که نشان می‌دهد یا $atoms\_of(c, x)$ درست است یا $atoms\_of(c', x)$ در هر دو حالت حکم اول در فرض استقرا وجود دارد اگر $atoms\_of(c, x)$ حکم دوم از فرض دوم استقرای اول و لم ۶۵.۴ به دست می‌آید و اگر $atoms\_of(c', x)$ حکم دوم از فرض دوم استقرای دوم و لم ۶۶.۴ به دست می‌آید.

□

قضیه ۸.۵ (قضیه‌ی درونیابی کریگ)**.** *برای هر دو مجموعه* Γ, Δ *که داشته باشیم:*

$$\Gamma \dashrightarrow \Delta$$

وجود دارد درونیابی به نام c *که داریم:*

$$\Gamma \dashrightarrow c$$

$$c \dashrightarrow \Delta$$

$$atoms\_incl\ c\ \Gamma\ \Delta$$

اثبات. از لم درونیابی ۷.۵ استفاده می‌کنیم و قرار می‌دهیم

$$G1 = \Gamma, G2 = \emptyset, D1 = \emptyset, D2 = \Delta$$

□

فصل ۶

اثبات قضیه‌ی درونیابی کریگ در منطق گزاره‌ای شهودی

۱.۶ دستگاه استنتاجی حساب رشته‌ها برای منطق گزاره‌ای شهودی

دستگاه حساب رشته‌ها برای منطق شهودی کاملا مشابه منطق کلاسیک است با این تفاوت که مجموعه زمینه تالی حداکثر می‌تواند یک عضو داشته‌باشد.

تذکر ۱.۶. *به وضوح قاعده rC را در اینجا نداریم.*

تذکر ۲.۶. *قاعده $l \supset$ تنها قاعده‌ای است که شاید ظاهرش مثل قاعده نظیر آن در منطق کلاسیک نباشد.*

تذکر ۳.۶. *برای نشان دادن حساب رشته‌های منطق شهودی از نماد $\mapsto$ استفاده می‌کنیم.*

تذکر ۴.۶. *برای منطق شهودی هم از نوع Node مشابه ۴.۵ استفاده می‌کنیم.*

$$\frac{}{A \mapsto A}\ \text{Axiom}$$

$$\frac{\Gamma \mapsto \Delta}{\Gamma, p \mapsto \Delta}\ \text{lW} \qquad \frac{\Gamma \mapsto}{\Gamma \mapsto p}\ \text{rW}$$

$$\frac{\Gamma, p, p \mapsto \Delta}{\Gamma, p \mapsto \Delta}\ \text{lC}$$

$$\frac{}{\bot \mapsto A}\ \bot$$

$$\frac{\Gamma, b \mapsto \Delta}{\Gamma, a \wedge b \mapsto \Delta}\ \wedge l2 \qquad \frac{\Gamma, a \mapsto \Delta}{\Gamma, a \wedge b \mapsto \Delta}\ \wedge l1$$

$$\frac{\Gamma \mapsto a \qquad \Gamma \mapsto b}{\Gamma \mapsto a \wedge b}\ \wedge r$$

$$\frac{\Gamma, a \mapsto \Delta \qquad \Gamma, b \mapsto \Delta}{\Gamma, a \vee b \mapsto \Delta}\ \vee l$$

$$\frac{\Gamma \mapsto b}{\Gamma \mapsto a \vee b}\ \vee r2 \qquad \frac{\Gamma \mapsto a}{\Gamma \mapsto a \vee b}\ \vee r1$$

$$\frac{\Gamma \mapsto a \qquad \Gamma, b \mapsto \Delta}{\Gamma, a \supset b \mapsto \Delta}\ \supset l$$

$$\frac{\Gamma, a \mapsto b}{\Gamma \mapsto a \supset b}\ \supset r$$

شکل ۱.۶: Sequent calculus formulation for propositional intuitionistic logic

تعریف ۵.۶. *در کد، حساب رشته‌ها برای منطق شهودی را یک نوع استقرایی تعریف می‌کنیم که یک Node می‌گیرد یک مجموعه مکرر می‌گیرد یک ترم از نوع option prop و یک گزاره خروجی می‌دهد.*

```
Inductive LKI: Node -> multiset -> option prop -> Prop :=
(* 1 *)
|LKIA : forall (p : prop), {{p}} ↦ Some p >< leaf
(* 2 *)
|LKIrW : forall G p n, G ↦ None >< n -> G ↦ Some p >< (⊙ n)
(* 3 *)
|LKIlW : forall G op p n, G ↦ op >< n -> G ∔̂ p ↦ op >< (⊙ n)
(* 4 *)
|LKIlC : forall G op p n, (G ∔̂ p) ∔̂ p ↦ op >< n -> G ∔̂ p ↦ op >< (⊙ n)
(* 5 *)
|LKIrA : forall G (a b : prop) m n,
G ↦ Some a >< n -> G ↦ Some b >< m -> G ↦ Some (a ∧ b) >< n ⩓ m
(* 6 *)
|LKIl1A : forall G op (a b : prop) n, (G ∔̂ a) ↦ op >< n ->
(G ∔̂ (a ∧ b)) ↦ op >< (⊙ n)
(* 7 *)
|LKIl2A : forall G op (a b : prop) n, (G ∔̂ b) ↦ op >< n ->
(G ∔̂ (a ∧ b)) ↦ op >< (⊙ n)
(* 8 *)
|LKIr1O : forall G (a b : prop) n, G ↦ Some a >< n
-> G ↦ Some (a ∨ b) >< (⊙ n)
(* 9 *)
|LKIr2O : forall G (a b : prop) n, G ↦ Some b >< n
-> G ↦ Some (a ∨ b) >< (⊙ n)
(* 10 *)
|LKIlO: forall G op (a b : prop) m n,
(G ∔̂ a) ↦ op >< n -> (G ∔̂ b) ↦ op >< m -> (G ∔̂ (a ∨ b)) ↦ op >< (m ⩓ n)
(* 11 *)
|LKIB : forall op, {{ ⊥ }} ↦ op >< leaf
(* 12 *)
|LKrI: forall G (a b : prop) n, (G ∔̂ a) ↦ Some b >< n
-> G  ↦ Some (a ⊃ b) >< (⊙ n)
(* 13 *)
```

```
|LKlI: forall G op (a b : prop) m n,
G ↦ Some a >< n -> (G ∔ b) ↦ op >< m -> (G ∔ (a ⊃ b)) ↦ op >< (m ≍ n)
where G ↦ p >< n := (LKI n G p).
```

۲.۶ اثبات قضیه‌ی درونیابی در منطق گزاره‌ای کلاسیک

لم ۶.۶ (لم درونیابی برای منطق شهودی)**.** *برای هر* $G1, G2, op$ *اگر داشته باشیم* $G1, G2 \mapsto op$ *آنگاه وجود دارد درونیاب* c *به طوری که:*

$$G1 \mapsto c \tag{۱.۶}$$

$$G2, c \mapsto op \tag{۲.۶}$$

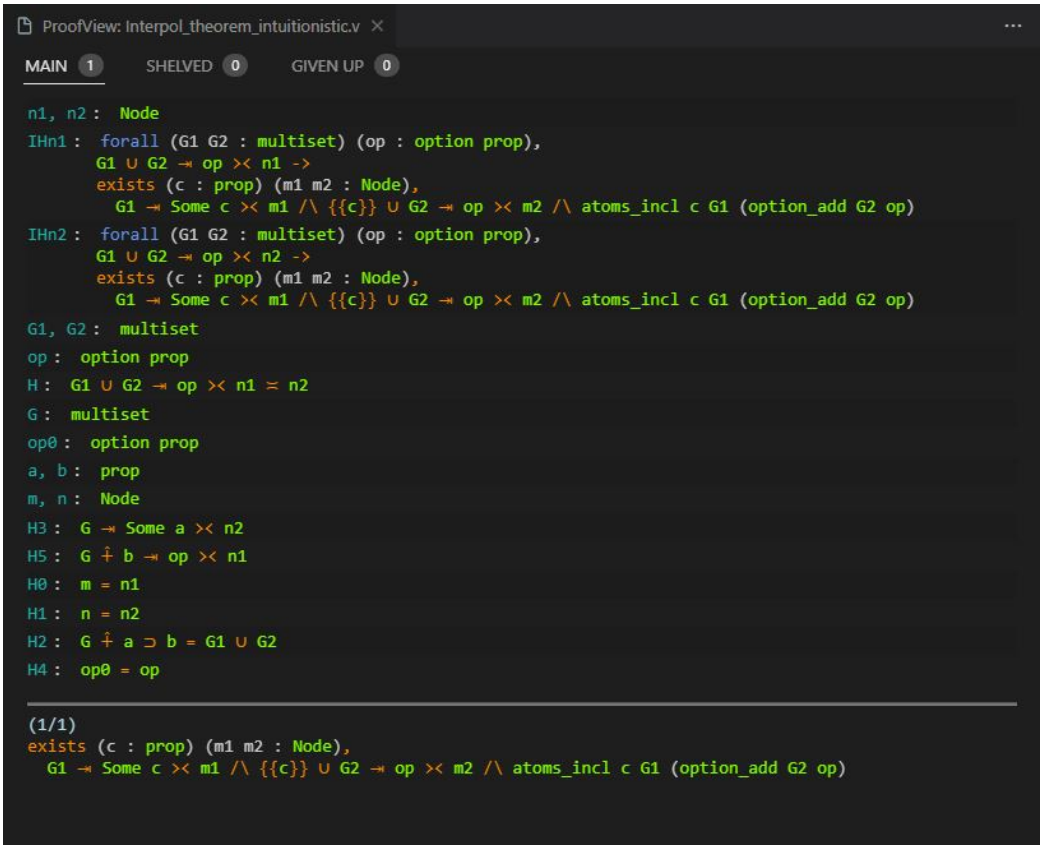
$$atoms\_incl\ c\ G1\ (option\_add(G2, op)) \tag{۳.۶}$$

```
Theorem LKI_Interpol_strong: forall n (G1 G2: multiset) (op : option prop),
G1 ∪ G2 ↦ op >< n ->
(exists (c : prop) m1 m2, G1 ↦ Some c >< m1 /\ {{c}} ∪ G2 ↦ op >< m2
/\ (atoms_incl c (G1) (option_add G2 op))).
```

اثبات. با احتساب حالت‌های کمتر، صوری‌شده‌ی اثبات این لم از معادل آن در منطق کلاسیک طولانی‌تر است؛ اما این امر به دلیل پیچیدگی‌های تکنیکی است که استفاده از نوع option به مسئله وارد می‌کند. به غیر از یک حالت بقیه اثبات و درونیاب‌ها عینا مشابه معادل آن حالت در منطق کلاسیک است، به همین دلیل از اثبات آن‌ها صرف نظر می‌کنیم و تنها آن یک حالت متفاوت را اثبات می‌کنیم.

اگر درخت اثبات به قاعده l $\supset$ ختم شده باشد داریم:

$$\supset l\ \frac{G1', G2', b \mapsto op \qquad G1', G2' \mapsto a}{G1', G2', a \supset b \mapsto op}$$



دو حالت وجود دارد:

۱. اگر $a \supset b \in G1$ آنگاه $\exists G', G' \hat{+} (a \supset b) = G1$ در فرض اول استقرا $(G' \hat{+} b), G2, op$ را قرار می‌دهیم و نتایج زیر به دست می‌آید:

$$\exists c,\ G', b \mapsto c$$

$$G2, c \mapsto op$$

$$atoms\_incl\ c\ (G' \hat{+} b)\ (option\_add(G2, op))$$

خلاقیت ویژه این حالت در استفاده از فرض استقرای دوم خود را نشان می‌دهد. در فرض استقرای دوم قرار می‌دهیم $G2, G', a$ و فرض استقرا به ما می‌دهد:

$$\exists c',\ G2 \mapsto c'$$

$$G', c' \mapsto a$$

$$atoms\_incl\ c'\ G2\ (option\_add(G', a))$$

درونیاب را $c' \supset c$ قرار می‌دهیم برای شرط ۱.۶ داریم:

$$\supset l\ \frac{\dfrac{G', c' \mapsto a \qquad \text{IW}\ \dfrac{G', b \mapsto c}{G', c', b \mapsto c}}{\supset r\ \dfrac{G', c', a \supset b \mapsto c}{G', a \supset b \mapsto c' \supset c}}}{G1 \mapsto c' \supset c}$$

برای شرط ۲.۶ داریم:

$$\supset l\ \frac{G2, c \mapsto op \qquad G2 \mapsto c'}{G2, c' \supset c \mapsto op}$$

برای اثبات شرط ۳.۶ باید اثبات کنیم برای هر x اگر $atoms\_of((c' \supset c), x)$ آنگاه حکم برقرار است. از لم ۶۹.۴ استفاده می‌کنیم پس یا $atoms\_of(c, x)$ درست است یا $atoms\_of(c', x)$ در حالت اول جفت حکم‌ها به‌راحتی از فرض استقرا و لم ۶۵.۴ اثبات می‌شوند. اگر حالت دوم برقرار باشد با لم ۷۴.۴ و حالت‌بندی روی فرض استقرا حکم ثابت می‌شود.

۲. اگر $a \supset b \in G2$ آنگاه $\exists G', G' \hat{+} (a \supset b) = G2$ در فرض اول استقرا $G1, (G' \hat{+} b), op$ را قرار می‌دهیم و نتایج زیر به دست می‌آید:

$$\exists c,\ G1 \mapsto c$$

$$G', b, c \mapsto op$$

$$atoms\_incl\ c\ G1\ (option\_add((G' \hat{+} b), op))$$

در فرض استقرای دوم قرار می‌دهیم $G1, G', a$ و فرض استقرا به ما می‌دهد:

$$\exists c',\ G1 \mapsto c'$$

$$G', c' \mapsto a$$

$$atoms\_incl\ c'\ G1\ (option\_add(G', a))$$

درونیاب را $c \wedge c'$ در نظر می‌گیریم و برای شرط ۱.۶ داریم:

$$\wedge r\ \frac{G1 \mapsto c' \qquad G1 \mapsto c}{G1 \mapsto c \wedge c'}$$

برای شرط ۲.۶ داریم:

$$\supset l\ \dfrac{\wedge l1\ \dfrac{G', c, b \mapsto op}{G', c \wedge c', b \mapsto op} \qquad \wedge l2\ \dfrac{G', c' \mapsto a}{G', c \wedge c' \mapsto a}}{\dfrac{G', c \wedge c', a \supset b \mapsto op}{G2, c \wedge c' \mapsto op}}$$

شرط ۳.۶ مشابه قبل اثبات می‌شود.

□

قضیه ۷.۶ (قضیه درونیابی کریگ برای منطق گزاره‌ای شهودی)**.** *برای هر مجموعه* Γ, Δ *که داشته باشیم:*

$$\Gamma \mapsto \Delta$$

وجود دارد درونیابی به نام c *که داریم:*

$$\Gamma \mapsto c$$

$$c \mapsto \Delta$$

$$atoms\_incl\ c\ \Gamma\ \Delta$$

اثبات. از لم درونیابی ۶.۶ استفاده می‌کنیم و قرار می‌دهیم

$$G1 = \Gamma, G2 = \emptyset, D2 = \Delta$$

□

فصل ۷

نتیجه‌گیری

برای پیاده‌سازی این اثبات روش‌های بسیار به آزمون گذارده شد و شکست خورد تا اینکه بتوان به روشی بهینه و کوتاه این اثبات را پیاده‌سازی کرد. اهمیت اصلی این پروژه به پیدا کردن روشی است که می‌توان با آن اثباتی از دید نظریه اثبات برای این قضیه ارائه داد، اینکار، هم راه را برای پیاده سازی اثبات‌های مشابه که از درخت اثبات به صورت ساختنی اثبات می‌شوند هموار می‌کند، هم اینکه کمک می‌کند تا اثبات‌های اشکال دیگر قضیه درونیابی به همین روش اثبات شوند، همانطور که دیدیم برای اثبات قضیه درونیابی در منطق شهودی بعد از تعریف دستگاه استنتاج با همان اصول قبلی اثبات به راحتی قابل تولید بود.

واژه‌نامه فارسی به انگلیسی

استلزام	implication
اثبات‌یار	proof assistant
حساب رشته‌ها	sequent calculus
حساب ساخت‌ها	calculus of constructions
درونیابی	interpolation
صوری‌سازی	formalization
سازنده	constructor
عطف	conjunction
فصل	disjunction
مجموعه زمینه تالی	succendent
مجموعه زمینه مقدم	anticedent
مجموعه‌ی مکرر	multiset
منطق خطی	linear logic
منطق وجهی	modal logic
نظریه انواع	type theory
نوع	type

واژه‌نامه انگلیسی به فارسی

anticedent	مجموعه زمینه مقدم
calculus of constructions	حساب ساخت‌ها
conjunction	عطف
constructor	سازنده
disjunction	فصل
formalization	صوری‌سازی
implication	استلزام
interpolation	درونیابی
linear logic	منطق خطی
modal logic	منطق وجهی
multiset	مجموعه‌ی مکرر
proof assistant	اثبات‌یار
sequent calculus	حساب رشته‌ها
succendent	مجموعه زمینه تالی
type	نوع
type theory	نظریه انواع

Bibliography

[1] Bruno Barras. "Sets in Coq, Coq in Sets". In: *Journal of Formalized Reasoning* 3.1 (Jan. 2010), pp. 29–48. DOI: 10.6092/issn.1972-5787/1695. URL: https://jfr.unibo.it/article/view/1695.

[2] Arthur Blot, Pierre-Évariste Dagand, and Julia Lawall. "From Sets to Bits in Coq". In: *Functional and Logic Programming - 13th International Symposium, FLOPS 2016, Kochi, Japan, March 4-6, 2016, Proceedings*. Ed. by Oleg Kiselyov and Andy King. Vol. 9613. Lecture Notes in Computer Science. Springer, 2016, pp. 12–28. DOI: 10.1007/978-3-319-29604-3_2. URL: https://doi.org/10.1007/978-3-319-29604-3%5C_2.

[3] Samuel R. Buss. "An Introduction to Proof Theory". In: *Handbook of Proof Theory, Studies in Logic and the Foundations of Mathematics*. Ed. by Samuel R. Buss. Amsterdam: Elsevier, 1998, p. 811.

[4] William Craig. "Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory". In: *The Journal of Symbolic Logic* 22.3 (1957), pp. 269–285. DOI: 10.2307/2963594.

[5] D.M. Gabbay and L. Maksimova. *Interpolation and Definability: Modal and Intuitionistic Logics*. Interpolation and Definability: Modal and Intuitionistic Logic. Clarendon Press, 2005. ISBN: 9780198511748. URL: https://books.google.nl/books?id=%5C_ysSDAAAQBAJ.

[6] Gerhard Gentzen. "Untersuchungen über das logische Schließen. I". In: *Mathematische Zeitschrift* 39 (1935), pp. 176–210. URL: https://api.semanticscholar.org/CorpusID:121546341.

[7] Helmut Pfeiffer. "Jean-Yves Girard. Proof theory and logical complexity. Volume I. Studies in proof theory, no. 1. Bibliopolis, Naples 1987, also distributed by Humanities Press, Atlantic Highlands, N.J.,

503 pp." In: *The Journal of Symbolic Logic* 54.4 (1989), pp. 1493–1494. DOI: 10.2307/2274839.

[8] Tom Ridge. "Craig's Interpolation Theorem formalised and mechanised in Isabelle/HOL". In: *CoRR* abs/cs/0607058 (2006). arXiv: cs/0607058. URL: http://arxiv.org/abs/cs/0607058.

[9] Carlos Simpson. *Set-theoretical mathematics in Coq*. 2004. arXiv: math/0402336 [math.LO].

[10] M. A. Taitslin. "Roger C. Lyndon. An interpolation theorem in the predicate calculus. Pacific journal of mathematics, vol. 9 (1959), pp. 129–142." In: *The Journal of Symbolic Logic* 25.3 (1960), pp. 273–274. DOI: 10.2307/2964711.

[11] G. Takeuti. *Proof Theory. Number 81 in Studies in Logic and the Foundations of Mathematics*. North-Holland, 1975.

[12] Robin Wilson and Charles Nash. "Four colours suffice: how the map problem was solved". In: *The Mathematical Intelligencer* 25 (Dec. 2003), pp. 80–83. DOI: 10.1007/BF02984867.

[13] Konrad Zdanowski. "On second order intuitionistic propositional logic without a universal quantifier". In: *J. Symb. Log.* 74 (Mar. 2009), pp. 157–167. DOI: 10.2178/jsl/1231082306.

Abstract

There are numerous interpolation theorems for different logics, as well as multiple semantical proofs (algebraic, kripke & lattic based) and syntactical proofs. Craig's interpolation theorem is one of these. One of the most beautiful and well-known proofs of interpolation is Maehara's demonstration of Craig's interpolation theorem. In this research, we establish the Maehara technique in propositional classical and intuitionistic logic using Coq proof assistant to prove Craig's interpolation theorem.

College of Science
School of Mathematics, Statistics, and Computer Science

Formalization of uniform interpolation theorem in IPC using Coq proof assistant

Asha Soroushpoor

Supervisor: Prof. Majid Alizadeh

A thesis submitted in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science

2023