Logica e Reti Logiche

(Episodio 9: Correttezza e completezza del metodo dei *tableaux* per la Logica del Primo Ordine)

Francesco Pasquale

20 aprile 2023

Nell'episodio precedente abbiamo visto che è sufficiente aggiungere due regole di estensione per applicare il metodo dei *tableaux*, che avevamo introdotto per la Logica Proposizionale, anche al caso della Logica del Primo Ordine.

In questo episodio dimostriamo, come avevamo fatto nell'Episodio 5 per il caso della Logica Proposizionale, che il metodo dei *tableaux* per la Logica del Primo Ordine è *corretto* e *completo*, ossia che tutte e sole le formule dimostrabili col metodo dei *tableaux* sono le formule valide.

Per comprendere il contenuto di questo episodio è necessario conoscere sintassi e semantica della logica del primo ordine (Episodio 7) e il metodo dei *tableaux* (Episodi 4 e 8); è anche utile aver già studiato le dimostrazioni di correttezza e completezza per il caso proposizionale (Episodio 5).

1 Formule soddisfacibili e correttezza del metodo

Per dimostrare la correttezza del metodo dei *tableaux* è utile dare un nome alle formule che hanno almeno una interpretazione in cui sono `T`.

Definizione 1.1 (Formule e insiemi di formule soddisfacibili)**.** Una formula $\mathcal{F}$ è *soddisfacibile* se esiste una interpretazione in cui $\mathcal{F}$ è `T`. Un insieme S di formule è *soddisfacibile* se esiste una interpretazione in cui tutte le formule di S sono `T`.

Esercizio 1. Osservare che una formula $\mathcal{F}$ è soddisfacibile se e solo se $\neg\mathcal{F}$ non è valida.

Esercizio 2. Fare un esempio di formula soddisfacibile e un esempio di formula non soddisfacibile. Per la formula soddisfacibile, esibire una interpretazione in cui è `T`, per quella non soddisfacibile dare una spiegazione intuitiva del perché non può esistere una interpretazione in cui è `T`.

Se I è una interpretazione in cui la formula $\mathcal{F}$ è `T` diremo anche che I *soddisfa* $\mathcal{F}$. Diremo che I soddisfa l'insieme S se I soddisfa tutte le formule in S.

Esercizio 3. Mostrare che l'insieme

$$S = \{\exists x[P(x) \wedge Q(x)],\ \forall x[P(x) \vee Q(x)],\ \neg\forall x P(x),\ \neg\forall x Q(x)\}$$

è soddisfacibile esibendo una interpretazione che soddisfa S.

Si ricordi che una formula $\mathcal{F}$ si dice *dimostrabile* col metodo dei *tableaux* se esiste un *tableau* chiuso che parte da $\neg\mathcal{F}$.

Teorema 1.2 (Correttezza)**.** Se una formula $\mathcal{F}$ è dimostrabile col metodo dei *tableaux* allora è valida.

Idea della dimostrazione. Per dimostrare che se $\mathcal{F}$ è dimostrabile col metodo dei *tableaux* allora $\mathcal{F}$ è valida, dimostreremo che se $\mathcal{F}$ non è valida allora $\mathcal{F}$ non è dimostrabile col metodo dei *tableaux*. Possiamo schematizzare la dimostrazione in questi tre passaggi:

1. Se $\mathcal{F}$ non è valida, allora $\neg\mathcal{F}$ deve essere soddisfacibile.
   [Segue direttamente dalle definizioni di formula valida e formula soddisfacibile].
2. Se $\neg\mathcal{F}$ è soddisfacibile, allora non è possibile ottenere un *tableau* chiuso a partire da $\neg\mathcal{F}$.
   [Questo è il passaggio cruciale. Spezziamolo in due passaggi.]
   2.1. Se ϑ è un ramo chiuso di un *tableau*, allora l'insieme S_ϑ, formato da tutte le formule sul ramo ϑ, non è soddisfacibile;
   [Se ϑ è chiuso allora in S_ϑ deve esserci sia una formula che la sua negata.]
   2.2. Se $\neg\mathcal{F}$ è soddisfacibile e uso le regole α, β, γ e δ per costruire un *tableau* a partire da $\neg\mathcal{F}$, allora nel *tableau* che costruisco ci sarà sempre un ramo ϑ tale che l'insieme S_ϑ, formato da tutte le formule sul ramo ϑ, è soddisfacibile.
   [Questo viene dal significato semantico delle regole di estensione del tableaux: lo dimostriamo nel successivo Lemma 1.3]
3. Quindi se ottengo un *tableau* chiuso a partire da $\neg\mathcal{F}$ allora $\mathcal{F}$ deve essere valida.
   [Segue direttamente dai punti 1 e 2.]

Ci manca un solo pezzo per completare il *puzzle*: dobbiamo dimostrare il punto (2.2).

Lemma 1.3. Sia S un insieme di formule soddisfacibile e sia $\mathcal{F}$ una formula in S. Se $\mathcal{F}$ è una

(a) α-formula, allora anche l'insieme $S \cup \{\alpha_1, \alpha_2\}$ è soddisfacibile;
(b) β-formula, allora almeno uno dei due insiemi $S \cup \{\beta_1\}$ e $S \cup \{\beta_2\}$ è soddisfacibile;
(c) γ-formula, allora anche l'insieme $S \cup \{\gamma(a)\}$ è soddisfacibile, quaunque sia il parametro a;
(d) δ-formula, allora anche l'insieme $S \cup \{\delta(a)\}$ è soddisfacibile, se a è un parametro che non compare già in qualche formula di S.

Esercizio 4. Riflettere sul fatto che il punto (2.2) della dimostrazione del Teorema 1.2 segue dal Lemma 1.3.

Esercizio 5. Dimostrare i punti (a) e (b) del Lemma 1.3.

Esercizio 6. Dimostrare il punto (c) del Lemma 1.3.

(Soluzione: L'insieme S è soddisfacibile per ipotesi, quindi deve esistere una interpretazione I che soddisfa S. In particolare, I soddisfa anche la nostra γ-formula $\mathcal{F}$. Quindi come costruiamo una interpretazione che soddisfa sia tutte le formule S sia la nuova formula $\gamma(a)$? Dobbiamo distinguere due casi: 1. Se il parametro "a" stava già in qualche formula in S, allora l'interpretazione I assegna già un elemento del dominio al parametro "a", ma in questo caso l'interpretazione I soddisfa anche la formula $\gamma(a)$ (perché I soddisfa γ e γ è di tipo UNIVERSALE*). 2. Se il parametro "a" non stava già in qualche formula in S, allora possiamo costruire una interpretazione I' che è uguale a I e in più assegna al parametro "a" un qualunque elemento del dominio. Quindi I' soddisfa tutte le formule in S e in più, siccome γ è di tipo* UNIVERSALE*, soddisfa anche $\gamma(a)$.)*

Esercizio 7. Dimostrare il punto (d) del Lemma 1.3.
(Suggerimento: L'insieme S è soddisfacibile per ipotesi, quindi deve esistere una interpretazione I che soddisfa S. In particolare una tale interpretazione I soddisfa la nostra δ-formula $\mathcal{F}$. Quindi come possiamo costruire una interpretazione I' che soddisfa tutte le formule in S e soddisfa anche la formula $\delta(a)$?)

2 *Tableaux* "sistematici", insiemi di Hintikka e completezza del metodo

Nella descrizione del metodo dei *tableaux* e negli esempi fatti finora abbiamo visto che c'è una certa "libertà d'azione" nella scelta di quale formula estendere in un determinato passaggio e in che modo farlo. Nel caso della logica proposizionale la scelta è limitata all'ordine con cui estendiamo le formule e gli effetti di questa arbitrarietà sono limitati al fatto che il *tableau* che otteniamo facendo delle scelte invece che altre può venirci più o meno lungo. Nel caso della logica del primo ordine però abbiamo visto che le formule di tipo UNIVERSALE possono anche essere usate più volte, con diversi parametri. Inoltre usare più volte le γ-formule e con i parametri opportuni è a volte necessario per ottenere un *tableau* chiuso. È possibile "automatizzare" le scelte per essere sicuri che, se è possibile ottenere un *tableau* chiuso da una data formula, prima o poi effettivamente lo otteniamo? Sì, è possibile.

Definizione 2.1 (*Tableaux* sistematici)**.** Chiamiamo *tableau* sistematico, un *tableau* ottenuto seguendo questo schema di estensione delle formule:

1. Fissiamo un ordinamento arbitrario dei parametri $a_1, a_2, a_3, \ldots$ che andremo a usare ogni volta che estendiamo una γ-formula o una δ-formula.
2. Estendiamo le formule nell'ordine in cui appaiono.
3. Ogni volta che estendiamo una γ-formula $\mathcal{F}$:
   - 3.1 Usiamo il primo parametro disponibile che non abbiamo già usato per estendere $\mathcal{F}$ sul ramo che stiamo considerando.
   - 3.2 Ripetiamo la formula $\mathcal{F}$ stessa sotto tutte le foglie del sottoalbero radicato in $\mathcal{F}$.

Facciamo un esempio. Prendiamo la formula $\exists y[P(y) \to \forall x P(x)]$, per la quale abbiamo già visto nell'episodio precedente che per ottenere un *tableau* bisogna usare due volte una γ-formula. Generiamo il *tableau* seguendo le indicazioni qui sopra.

$$
\begin{array}{ccc}
\neg\exists y[P(y) \to \forall x P(x)] & (1) & \\
\neg[P(a) \to \forall x P(x)] & (2) & \\
\neg\exists y[P(y) \to \forall x P(x)] & (3) & a \\
P(a) & (4) & \\
\neg\forall x P(x) & (5) & \\
\neg[P(b) \to \forall x P(x)] & (6) & \\
\neg\exists y[P(y) \to \forall x P(x)] & (7) & b \\
\neg P(c) & (8) & \\
P(b) & (9) & \\
\neg\forall x P(x) & (10) & \\
\neg[P(c) \to \forall x P(x)] & (11) & \\
\neg\exists y[P(y) \to \forall x P(x)] & (12) & c \\
\neg P(d) & (13) & \\
P(c) & (14) & \\
\neg\forall x P(x) & (15) &
\end{array}
$$

1. Siccome (1) è γ-formula, la estendiamo usando il primo parametro disponibile, "a", ottenendo la formula (2) e la riaggiungiamo sotto come formula (3), ricordandoci che "a" è l'ultimo parametro con cui l'abbiamo estesa.
2. Dalla (2) seguono (4) e (5).
3. Dalla (3), che estendiamo con il parametro disponibile successivo, "b", otteniamo la (6). Siccome (3) è γ-formula, la riaggiungiamo di nuovo come (7).
4. La (4) è già atomica.
5. Dalla (5) otteniamo la (8). Osservate che, siccome la (5) è δ-formula, devo usare un parametro mai usato prima, quindi prendo il primo disponibile, cioè c.
6. Dalla (6) seguono (9) e (10).
7. La (7) è di nuovo la formula iniziale. La estendo con il primo parametro con cui non l'ho già estesa, "c", ottenendo la (11), e la riporto in fondo come formula (12).
8. La (8) è atomica.
9. La (9) è atomica.
10. Dalla (10) ottengo la (13).
11. Dalla (11) seguono (14) e (15) e finalmente possiamo fermarci perché la (14) e la (8) sono in contraddizione.

Come vedete, la procedura sistematica ci ha fatto ottenere un *tableau* molto più lungo di quello che avevamo ottenuto nell'episodio precedente facendo delle scelte "ragionate" ogni volta che ne avevamo l'opportunità. Ma ci ha condotto ad un *tableau* chiuso in modo "automatico".

Esercizio 8. Generare un *tableau* sistematico per ognuna delle formule nell'Esercizio 5 dell'Episodio 6.

Oltre ai *tableaux* sistematici, per procedere alla dimostrazione di completezza del metodo abbiamo bisogno di un altro ingrediente, che avevamo già usato anche nel caso della logica proposizionale.

Definizione 2.2 (Insiemi di Hintikka)**.** Un insieme di formule S per cui valgono le cinque proprietà seguenti si chiama insieme di Hintikka:

H_0 : S non contiene sia una formula $\mathcal{F}$ che la sua negata $\neg\mathcal{F}$;
H_1 : Se S contiene una α-formula, allora S contiene anche entrambe le sue componenti α_1 e α_2;
H_2 : Se S contiene una β-formula allora S contiene anche almeno una delle sue componenti β_1 e β_2.
H_3 : Se S contiene una γ-formula, allora S contiene anche le formule $\gamma(a_1), \gamma(a_2), \gamma(a_3), \ldots,$ per tutti gli infiniti parametri $a_1, a_2, a_3, \ldots$.
H_4 : Se S contiene una δ-formula, allora S contiene anche la formula $\delta(a)$, per almeno uno dei parametri a.

Osservate che le proprietà H_0, H_1 e H_2 sono le stesse che avevamo nel caso della logica proposizionale. Le proprietà H_3 e H_4 estendono la definizione di insieme di Hintikka a insiemi che contengono anche formule della logica del primo ordine.

Ora possiamo dare uno schema della dimostrazione del teorema di completezza, sulla falsariga di quella del teorema di correttezza.

Teorema 2.3 (Completezza)**.** Se una formula $\mathcal{F}$ è valida allora è dimostrabile col metodo dei *tableaux*.

Idea della dimostrazione: Possiamo schematizzare la dimostrazione in questi passaggi.

1. Se un *tableau* sistematico che parte da $\neg\mathcal{F}$ non si chiude allora deve avere almeno un ramo aperto (eventualmente infinitamente lungo)
   [Per la definizione di tableau chiuso.]

2. Se ϑ è un ramo aperto (eventualmente infinitamente lungo) di un *tableau* sistematico allora l'insieme S_ϑ (eventualmente infinito) delle formule sul ramo ϑ è soddisfacibile.
   [Questo è il passaggio cruciale. Spezziamolo in due passaggi.]

   2.1 Se ϑ è un ramo aperto di un *tableau* sistematico, allora S_ϑ è un insieme di Hintikka.
   [Segue dalle definizioni di tableau sistematico e di insieme di Hintikka.]

   2.2 Un insieme di Hintikka è soddisfacibile.
   [Questo segue dalle proprietà di un insieme di Hintikka: lo dimostriamo nel successivo Lemma 2.4]

3. Quindi se eseguo un *tableau* sistematico partendo da $\neg\mathcal{F}$ prima o poi devo chiudere tutti i rami.
   [Segue dai punti 1 e 2 e dal fatto che se $\mathcal{F}$ è valida allora $\neg\mathcal{F}$ non è soddisfacibile]

Anche qui, per completare il *puzzle* ci resta solo da dimostrare il punto (2.2), che si chiama Lemma di Hintikka.

Lemma 2.4 (Lemma di Hintikka)**.** Se S è un insieme di Hintikka, allora S è soddisfacibile.

Esercizio 9. Dimostrare il Lemma di Hintikka.
(Suggerimento: Per induzione sul numero di connettivi e quantificatori della formula. Base: Tutte le formule che contengono al più un connettivo sono atomiche quindi, per la proprietà H_0, il sottoinsieme delle formule con al più un connettivo è soddisfacibile. Passo induttivo: Fissato un $n \in \mathbb{N}$, supponiamo che il sottoinsieme di formule che contengono al più n connettivi e quantificatori è soddisfacibile e dimostraiamo che anche il sottoinsieme formato dalle formule che contengono al più $n+1$ connettivi e quantificatori è soddisfacibile.)

Esercizio 10. Per ognuna delle formule seguenti, dire se la formula è vadida oppure no. In caso affermativo dimostrarla con il metodo dei *tableaux*, in caso negativo esibire una interpretazione in cui la formula è falsa

1. $\exists x\, P(x) \to \forall x P(x)$
2. $\forall x[P(x) \vee Q(x)] \to \forall x P(x) \vee \forall x Q(x)$
3. $\forall x \forall y \forall z[P(x,x) \wedge [P(x,z) \to P(x,y) \vee P(y,z)]] \to \exists y \forall z P(y,z)$
4. $\forall x \exists y P(x,y) \wedge \exists x \forall y Q(x,y) \to \exists x \exists y\, [P(x,y) \wedge Q(x,y)]$
5. $\exists x \forall y P(x,y) \wedge \exists x \forall y Q(x,y) \to \forall y\, (\exists x P(x,y) \wedge \exists x Q(x,y))$

3 Conclusioni

In questo episodio abbiamo dimostrato correttezza e completezza del metodo dei *tableaux* per la logica del primo ordine. Nel percorso, abbiamo introdotto una procedura sistematica per eseguire i *tableaux*. Si osservi che con la procedura sistematica siamo sicuri che, se esiste un *tableau* chiuso che parte da una certa formula, *prima o poi* la troviamo, ma nessuno ci dice *quanto dobbiamo andare avanti* per trovarla. . .

In questa prima parte del corso abbiamo studiato i fondamenti della logica proposizionale e della logica del primo ordine. Nella seconda parte del corso vedremo come la logica proposizionale può essere implementata nei *circuiti* per fargli eseguire operazioni più o meno complesse.

Per quanto riguarda la logica del primo ordine, da un lato la reincontrerete direttamente quando studierete intelligenza artificiale, dall'altro avere una certa dimestichezza con il ruolo dei quantificatori sarà essenziale per comprendere i corsi teorici che incontrerete al secondo anno, quali per esempio *Fondamenti di informatica* e *Algoritmi*.

Bonus Track

In questa *bonus track* vediamo, brevemente e informalmente, in che modo il linguaggio della logica del primo ordine viene utilizzato per fornire fondamenta rigorose alle teorie matematiche e accenniamo alle implicazioni e ai limiti di questo approccio.

A Teorie e modelli

Nell'Episodio 6 abbiamo introdotto un sistema assiomatico per la logica proposizionale, costituito dai seguenti schemi di assiomi

$$\begin{aligned}
A_1: &\quad X \to (Y \to X)\\
A_2: &\quad (X \to (Y \to Z)) \to ((X \to Y) \to (X \to Z))\\
A_3: &\quad (\neg X \to \neg Y) \to ((\neg X \to Y) \to X)
\end{aligned}$$

e dalla regola di inferenza Modus Ponens

$$\frac{X, X \to Y}{Y}$$

Come il metodo dei *tableaux*, anche i sistemi assiomatici si possono estendere alla logica del primo ordine aggiungendo alcuni schemi di assiomi e regole di inferenza. Per esempio, si può dimostrare che è sufficiente aggiungere i due schemi di assiomi

$$\begin{aligned}
A_4: &\quad \forall x\varphi(x) \to \varphi(a)\\
A_5: &\quad \varphi(a) \to \exists x\varphi(x)
\end{aligned}$$

e le due regole di inferenza

$$\frac{\varphi(a) \to X}{\exists x\varphi(x) \to X} \qquad\qquad \frac{X \to \varphi(a)}{X \to \forall x\varphi(x)} \tag{1}$$

(dove X è chiusa e "a" è un parametro che non compare in X o in φ) per ottenere un sistema assiomatico corretto e completo per la logica del primo ordine.

Esercizio 11. Verificare che i due schemi di assiomi A_4 e A_5 sono formule valide.

Esercizio 12. Verificare che le due regole di inferenza in (1) sono corrette (ci fanno passare da formule valide a formule valide).

Una *teoria del primo ordine* è un sistema formale in cui, oltre agli schemi di assiomi logici (per esempio, gli schemi $A1 - A5$ che abbiamo riportato qui) compaiono degli "assiomi propri" della teoria, cioè delle formule che non sono valide in generale, ma che caratterizzano le formule che si vogliono includere nella teoria. I simboli usati negli assiomi propri costituiscono il *linguaggio* della teoria.

Per esempio, consideriamo la teoria in cui c'è una sola lettera proposizionale P e i tre assiomi seguenti:

$$\begin{aligned} B_1 &: \quad \forall x P(x,x) \\ B_2 &: \quad \forall x \forall y [P(x,y) \to P(y,x)] \\ B_3 &: \quad \forall x \forall y \forall x [P(x,y) \to (P(y,z) \to P(x,z))] \end{aligned}$$

Si osservi che ogni interpretazione che soddisfi tutti e tre gli assiomi qui sopra deve assegnare alla lettera predicativa P una relazione che sia *riflessiva* (assioma B_1), *simmetrica* (assioma B_2) e *transitiva* (assioma B_3). In altri termini, deve essere una *relazione* di equivalenza.

Una interpretazione in cui tutti gli assiomi propri di una teoria sono `T` si chiama *modello* della teoria.

Esercizio 13. Dare un modello per la teoria definita dagli assiomi $B_1 - B_3$.

Possiamo anche dire che una teoria è l'insieme di tutte le formule che si possono ottenere dagli assiomi propri usando le regole di inferenza. Si osservi che, se le regole di inferenza sono "corrette", ogni interpretazione che soddisfa tutti gli assiomi propri di una teoria deve soddisfare anche tutte le formule della teoria.

Diciamo che una teoria è *soddisfacibile* se ammette almeno un modello.

Esercizio 14. Dare un esempio di teoria non soddisfacibile.

Gli assiomi propri di una teoria si dicono *indipendenti* se non è possibile ottenere uno degli assiomi dagli altri utilizzando le regole di inferenza.

Esercizio 15. Esibire

1. Una interpretazione I_1 che soddisfi gli assiomi B_1 e B_2, ma non l'assioma B_3;
2. Una interpretazione I_2 che soddisfi gli assiomi B_2 e B_3, ma non l'assioma B_1;
3. Una interpretazione I_2 che soddisfi gli assiomi B_1 e B_3, ma non l'assioma B_2.

Concludere che gli assiomi B_1, B_2, B_3 devono essere indipendenti.

In molte teorie è necessario utilizzare il concetto di "uguaglianza". Questo si può fare o considerandola un elemento logico oppure definendo l'uguaglianza stessa tramite degli assiomi: *riflessività* e *sostitutività*

$$\begin{aligned} U_1 &: \quad \forall x P(x,x) && \text{(riflessività)} \\ U_2 &: \quad P(x,y) \to [\phi(x,x) \to \phi(x,y)] && \text{(sostitutività)} \end{aligned}$$

dove ϕ è una formula qualunque.

Esempio. L'aritmetica di Peano può essere formalizzata usando

- Una lettera predicativa P con due argomenti (l'*uguaglianza*, scriveremo $x = y$ invece che $P(x,y)$ che $x \neq y$ invece che $\neg P(x,y)$)
- Una costante "a" (lo *zero*, scriveremo 0 invece che a)
- Tre lettere funzionali f_1, f_2, f_3, la prima con un solo argomento (il *successivo*, scriveremo x' invece che $f_1(x)$) e le altre due con due argomenti (*somma* e *moltiplicazione*, scriveremo $x + y$ invece che $f_2(x,y)$ e $x \cdot y$ invece che $f_3(x,y)$)

$S_1 : \quad \forall x \forall y \forall z [x = y \to (x = z \to y = z)]$
$S_2 : \quad \forall x \forall y [x = y \to x' = y']$
$S_3 : \quad \forall x [0 \neq x']$
$S_4 : \quad \forall x \forall y [x' = y' \to x = y]$
$S_5 : \quad \forall x [x + 0 = x]$
$S_6 : \quad \forall x \forall y [x + y' = (x + y)']$
$S_7 : \quad \forall x [x \cdot 0 = 0]$
$S_8 : \quad \forall x \forall y [x \cdot y' = x \cdot y + x]$
$S_9 : \quad \varphi(0) \to [\forall x (\varphi(x) \to \varphi(x')) \to \forall x \varphi(x)]$ (dove φ è una formula qualunque)

Si noti che $S_1 - S_8$ sono *assiomi*, mentre S_9 è uno *schema di assioma* (alla φ possiamo sostituire una formula qualunque)

Esercizio 16. Verificare che i numeri naturali $\mathbb{N}$ con il concetto di *successivo* e le operazioni di *somma* e *prodotto* sono un modello della teoria definita da $S_1 - S_9$.

Esercizio 17. Perché invece i numeri interi $\mathbb{Z}$ non sono un modello per questa teoria?

B Consistenza, completezza sintattica e decidibilità

Visto che gli assiomi propri di una teoria non sono validi in generale (possono essere `T` in alcune interpretazioni e `F` in altre) potrebbe succedere che in una certa teoria sia possibile dimostrare (derivare dagli assiomi propri usando le regole di inferenza) sia una formula $\mathcal{F}$ sia la sua negata $\neg\mathcal{F}$. In questo caso la teoria si dice *inconsistente*. Si dice *consistente* una teoria in cui questo non può succedere.

Una teoria si dice *sintatticamente completa* se, data una qualunque formula $\mathcal{F}$ scritta nel linguaggio della teoria, o si può dimostrare $\mathcal{F}$ oppure si può dimostrare $\neg\mathcal{F}$.

Consistenza e completezza sintattica sono due proprietà che sarebbero desiderabili per qualunque teoria matematica. Tuttavia, negli anni trenta Gödel ha dimostrato un teorema che dice, a grandi linee, che ogni teoria "sufficientemente complessa" (inclusa l'aritmetica di Peano) non può essere contemporaneamente consistente e sitatticamente completa.

Teorie più "semplici" possono essere sia consistenti che complete. Per esempio, se consideriamo soltanto di assiomi da S_1 a S_6 e lo schema di assioma S_9 della sezione precedente (essenzialmente, se togliamo gli assiomi che caratterizzano la moltiplicazione nell'aritmetica di Peano) otteniamo l'aritmetica di Presburger, che si può dimostrare essere sia consistente che sintatticamente completa.

Esercizio 18. Osservare che la logica del primo ordine (ossia la teoria del primo ordine senza assiomi propri) è consistente, ma non è sintatticamente completa.

Un altro concetto utile da menzionare è la *decidibilità*. Una teoria si dice *decidibile* se esiste un algoritmo che prende in input una formula $\mathcal{F}$ e restituisce SI se la formula è un teorema e restituisce NO se $\mathcal{F}$ non è un teorema. Una teoria di dice *semi-decidibile* se esiste un algoritmo che restituisce SI se la formula è un teorema e restituisce NO oppure non termina se non è un teorema.

Esercizio 19. Osservare che la logica del primo ordine (ossia la teoria del primo ordine senza assiomi propri) è semi-decidibile (si può dimostrare che non è decidibile).

C Oltre la logica del primo ordine

Cosa c'è oltre la logica del primo ordine? Prima di tutto le logiche di "ordine superiore" (per esempio, nella logica del *secondo* ordine i quantificatori possono essere applicati anche ai predicati, oltre che alle variabili). Poi ci sono altri tipi di "ragionamenti logici" per i quali è necessario introdurre i concetti di "conoscenza" e "tempo". Vediamo brevemente un esempio che potrebbe darvi un'idea di cosa si tratta.

Esercizio 20. Su un isola ci sono 1000 abitanti: 900 con gli occhi marroni e 100 con gli occhi blu. Ogni abitante conosce il colore degli occhi degli altri abitanti, ma non conosce il colore dei propri occhi. Sull'isola vige una regola ferrea che tutti seguono alla lettera: se un abitante scopre il colore dei propri occhi deve andare via dall'isola alla prima occasione con una nave che arriva sull'isola tutti i giorni a mezzogiorno e riparte poco dopo.

Un giorno sull'isola arriva un forestiero con gli occhi blu che, al momento di ripartire, saluta sulla piazza tutti gli abitanti dicendo "È stato bello incontrare su quest'isola qualcuno con gli occhi blu". Secondo voi che cosa succede sull'isola?

(In prima battuta si potrebbe osservare che la frase del forestiero non aggiunge nulla a ciò che ogni abitante conosce già, quindi non dovrebbe succedere nulla. Invece, per i primi novantanove giorni non succede niente, ma al centesimo giorno tutti gli abitanti con gli occhi blu lasciano l'isola. Come si spiega? Vediamo.
Immaginiamo che gli abitanti con gli occhi blu siano n invece che 100.
Se $n = 1$, allora chiaramente l'unico abitante con gli occhi blu, chiamiamolo B_1, vedendo che nessuno sull'isola ha gli occhi blu, con la frase del forestiero scoprirebbe di essere lui ad avere gli occhi blu. Quindi il giorno 1 l'abitante B_1 lascerebbe l'isola.
Se $n = 2$, ognuno dei due abitanti con gli occhi blu, chiamiamoli B_1 e B_2, vedrebbe sull'isola un solo abitante con gli occhi blu. L'abitante B_2 quindi saprebbe che, se B_1 fosse l'unico abitante con gli occhi blu (il caso $n = 1$) lascerebbe l'isola il giorno 1. Non vedendo partire B_1 il giorno 1, B_2 ne dedurrebbe che anche i suoi occhi sono blu. Chiaramente B_1 farebbe lo stesso ragionamento a parti invertite con B_2 e quindi il giorno 2 entrambi gli abitanti B_1 e B_2 saprebbero il colore dei loro occhi e lascerebbero l'isola.
Se $n = 3$, ognuno dei tre abitanti con gli occhi blu, chiamiamoli B_1, B_2 e B_3, vedrebbe sull'isola due soli abitanti con gli occhi blu...
....
Se $n = 100$, ...
Ma allora che cosa ha aggiunto la frase del forestiero a quello che gli abitanti già sapevano? Per capirlo, è utile partire dal caso $n = 2$: in quel caso è vero che già prima della frase del forestiero sia B_1 che B_2 sapevano che sull'isola c'era qualcuno con gli occhi blu, ma B_2 non sapeva se B_1 sapeva che sull'isola c'era qualcuno con gli occhi blu. Dopo la frase del forestiero B_2 sa che B_1 sa che sull'isola c'è qualcuno con gli occhi blu (lo stesso vale per B_1: prima della frase del forestiero non sapeva se B_2 sapeva). In generale, si dice che un fatto F è "common knowledge" se
- Tutti sanno F;
- Tutti sanno che tutti sanno F;
- Tutti sanno che tutti sanno che tutti sanno F;
e così via fino all'infinito. L'affermazione del forestiero ha reso "common knowledge" il fatto che esiste sull'isola qualcuno con gli occhi blu.)