Model checking basato su Tableau

Stefano Boschi

Metodi e strumenti per l'analisi e la verifica

Introduzione

- Nel model checking classico gli algoritmi di verifica sono tipicamente basati sulla ricerca esaustiva nello **spazio degli stati** di un modello di un sistema;
- Il sistema viene rappresentato per intero all'interno del calcolatore $\Rightarrow$ Il model checker "controlla" poi tutti i possibili comportamenti del sistema a spazio degli stati finito $\Rightarrow$ Ogni stato del modello è etichettato e coinvolto nell'analisi della formula;
- Questo approccio fa si che il model checking sia vulnerabile ad un problema pratico:

  Il numero degli stati può eccedere l'ammontare di memoria a disposizione;

- Questo problema è conosciuto come Esplosione dello spazio degli stati;
- Sono stati sviluppati vari metodi per combattere questo problema, uno di essi è il model checking basato su Tableau;

Model checking basato su Tableau

- I metodi basati su Tableau appartengono alla "famiglia" degli algoritmi **On the fly**, nei quali lo stato globale dell'automa non viene costruito completamente prima di applicare l'algoritmo di etichettamento, ma vengono via via generate solo le regioni dell'automa che sono strettamente necessarie a verificare la formula;

- Nel caso dei metodi basati su Tableau si prende uno stato s di un modello M, una proprietà ϕ e si cerca di determinare se:

$$s \models^M \phi$$

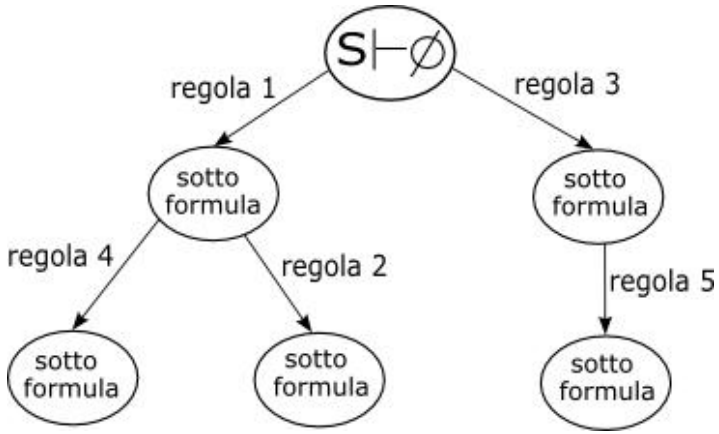
da questa, gli algoritmi basati su Tableau effettuano una ricerca nello **spazio degli stati** accessibile da $s \Rightarrow$ Lo spazio degli stati (un albero) viene costruito in modo incrementale, decomponendo la formula data in sotto-formule.

$$\Downarrow$$

L'albero generato è costituito solo dai nodi necessari a controllare la formula $\Rightarrow$ Non si memorizza l'automa completo $\Rightarrow$ Risparmio di spazio in memoria;

Costruzione del Tableau

Per verificare se $s \models^M \phi$ si prende la formula $s \vdash \phi$ e utilizziamo alcune **regole di inferenza** per generare una ricerca della prova, cioè un albero (un **Tableau**).



- La costruzione dell'albero procede, in ogni ramo, finchè non si arriva ad un nodo a cui non si può applicare nessuna regola di inferenza, questo nodo sarà una *foglia* dell'albero;
- Le regole di inferenza non sono ambigue $\Rightarrow$ Il tableau è generato in modo automatico;
- Per determinare se un certo stato soddisfa la formula data si deve guardare la "forma" delle foglie del tableau prodotto;

Le regole di inferenza

La costruzione dell'albero avviene tramite le seguenti *regole di inferenza*:

$$1)\ \frac{s \vdash_\Delta \phi_1 \wedge \phi_2}{s \vdash_\Delta \phi_1 \quad s \vdash_\Delta \phi_2}$$

$$2.a)\ \frac{s \vdash_\Delta \phi_1 \vee \phi_2}{s \vdash_\Delta \phi_1} \qquad 2.b)\ \frac{s \vdash_\Delta \phi_1 \vee \phi_2}{s \vdash_\Delta \phi_2}$$

$$3)\ \frac{s \vdash_\Delta [a]\phi}{s_1 \vdash_\Delta \phi_1 \ \ldots \ s_n \vdash_\Delta \phi_n}\ \text{se}\ \{s_1, \ldots, s_n\} = \left\{ s' | s \xrightarrow{a} s' \right\}$$

$$4)\ \frac{s \vdash_\Delta < a > \phi}{s' \vdash_\Delta \phi}\ \text{se}\ s \xrightarrow{a} s'$$

Queste formule sono relative all'*Hennessy-Milner logic*. Il simbolo Δ rappresenta l'*ambiente*, che rappresenta una piccola memoria usata dall'algoritmo durante la costruzione dell'albero: Nella logica *HML* non viene utilizzato, è utile nelle logiche in cui è presente il punto fisso, come il $\mu - calcolo$.

Soddisfacibilità

Per verificare che $s \models^M \phi$ si deve guadare come sono etichettate le foglie del Tableau;

Si dice che una *foglia ha successo* se è in una delle seguenti forme:

$$i)\ \ s \vdash_\Delta True$$

$$ii)\ \ s \vdash_\Delta [a]\phi$$

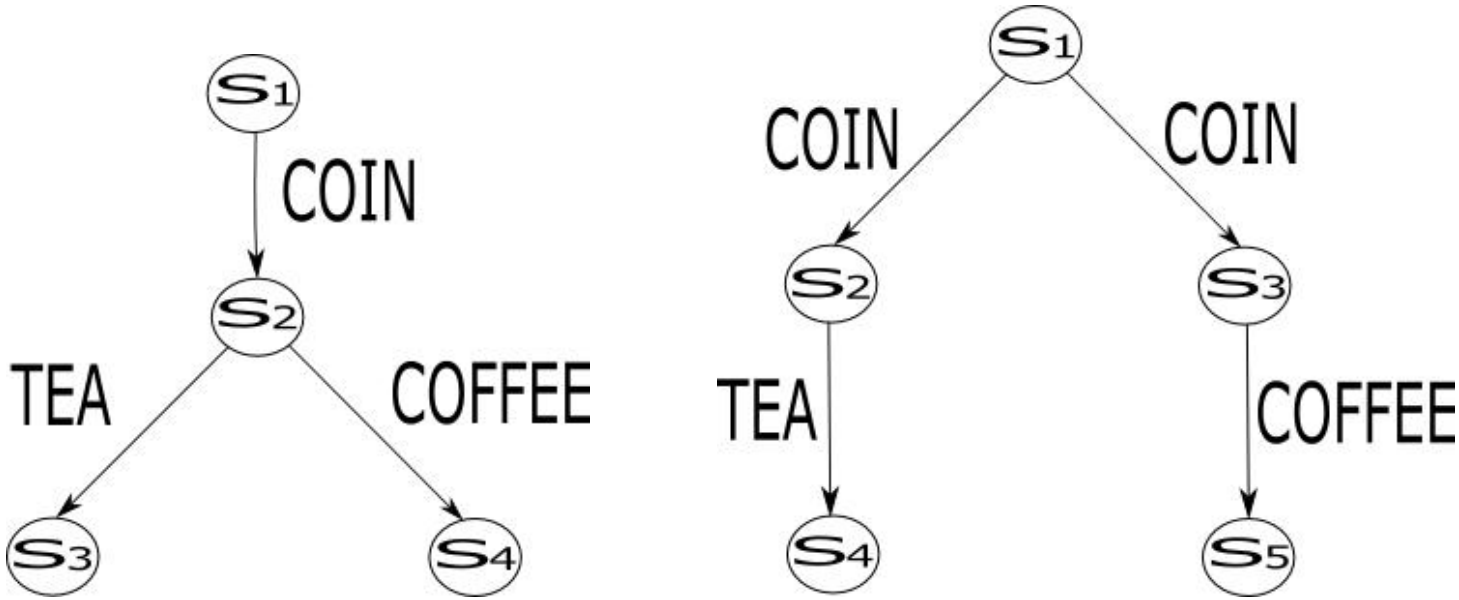
La forma *ii*) implica che non esiste una transizione etichettata con a da s.

Un **Tableau ha successo** se *tutte le sue foglie hanno successo*.

- Se esiste un Tableau che ha successo per la formula $s \vdash_\Delta \phi$ allora $s \models^M \phi$;
- Viceversa, se non esiste un tableau che ha successo, concludiamo che $s \not\models^M \phi$.

Esempio

Prendiamo due modelli (descritti attraverso un *LTS*) che descrivono una macchinetta del caffè:



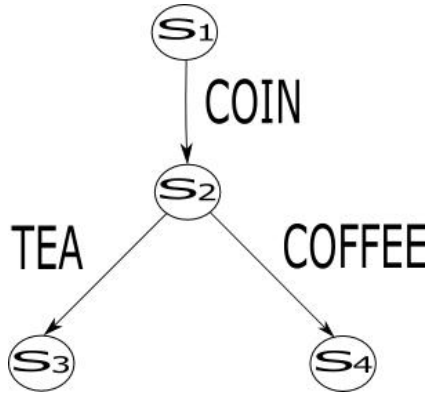
Determiniamo, attraverso la costruzione di un tableau, se gli stati s_1 dei due modelli soddisfano la seguente formula:

$$[COIN](< TEA > True \wedge < COFFEE > True)$$

La formula dice: "Dopo aver inserito una moneta si può scegliere tra il caffé e il thé".

Esempio

Primo modello



$$s_1 \vdash_\Delta [COIN](< TEA > True \wedge < COFFEE > True)$$

$$\downarrow$$

$$s_2 \vdash_\Delta < TEA > True \wedge < COFFEE > True$$

$$\swarrow \quad \searrow$$

$$s_2 \vdash_\Delta < TEA > True \qquad s_2 \vdash_\Delta < COFFEE > True$$

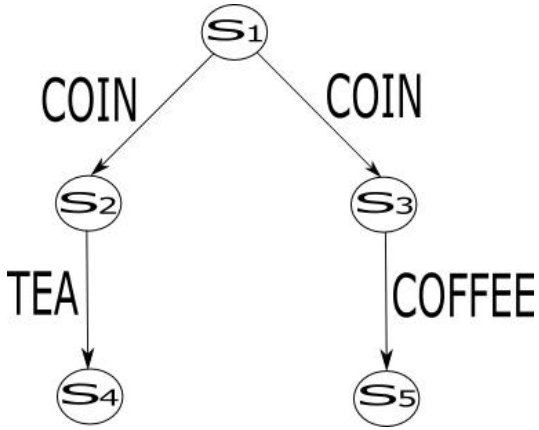
$$\downarrow \qquad\qquad \downarrow$$

$$s_3 \vdash_\Delta True \qquad s_4 \vdash_\Delta True$$

Tutte le foglie sono nella forma $s \vdash_\Delta True$, quindi il Tableau ha successo per la formula data $\Rightarrow$ Lo stato s_1 soddisfa la formula in esame.

Esempio

Secondo modello



$$s_1 \vdash_\Delta [COIN](< TEA > True \wedge < COFFEE > True)$$

$$s_2 \vdash_\Delta < TEA > True \wedge < COFFEE > True \qquad s_3 \vdash_\Delta < TEA > True \wedge < COFFEE > True$$

$$s_2 \vdash_\Delta < TEA > True \qquad s_2 \vdash_\Delta < COFFEE > True \qquad s_3 \vdash_\Delta < TEA > True \qquad s_3 \vdash_\Delta < COFFEE > True$$

$$s_2 \vdash_\Delta True \qquad s_3 \vdash_\Delta True$$

Non tutte le foglie sono nella forma presentata in $s \vdash_\Delta True$ o $s \vdash_\Delta [a]\phi$, quindi si può affermare che lo stato s_1 non soddisfa la formula data.

Tableau & μ – *calcolo*

- Il μ – *calcolo* è un'estensione dell'*Hennessy-Milner logic* in cui sono presenti gli **operatori di punto fisso** $\Rightarrow$ Si possono trattare "proprietà infinite";

- Uno degli obbiettivi dei metodi basati su Tableau è quello di utilizzare la minima quantità di memoria possibile $\Rightarrow$ Questa caratteristica deve essere mantenuta anche quando si trattano i punti fissi;

- Un aiuto ci viene dato dalla semantica dei punti fissi $\Rightarrow$ Si deve supporre di analizzare il "percorso" del punto fisso con le regole scompattate, finchè non si arriva al *bottom* (min-pf), o al *top* (max-pf), dello stesso, ottenendo quindi una formula di questo tipo:

$$s \vdash_{\Delta} X_0 \qquad \text{dove: } X_0 = True|False$$

- Chiaramente una formula di questo tipo è una foglia del Tableau $\Rightarrow$ Se la foglia ha successo o meno dipende dal punto fisso analizzato;

Tableau & μ — *calcolo*

Minimo e massimo punto fisso

Per l'operatore di **minimo punto fisso** si ha un percorso da $s \vdash_\Delta \mu X.\phi_X$ a $s \vdash_\Delta X_0$ e, dato che :

$$s \models^M \mu X.\phi_X \quad sse \quad s \models^M \bigvee_{i \geq 0} X_i \ , \ dove \ \begin{array}{l} X_0 = False \\ X_{i+1} = \phi_{X_i} \end{array}$$

si ottiene la formula $s \vdash_\Delta False$, quindi tutto quel percorso non ha successo.

Dualmente, per l'operatore di **massimo punto fisso** si ha un percoso da $s \vdash_\Delta \nu X.\phi_X$ a $s \vdash_\Delta X_0$ e, visto che:

$$s \models^M \nu X.\phi_X \quad sse \quad s \models^M \bigwedge_{i \geq 0} X_i \ , \ dove \ \begin{array}{l} X_0 = True \\ X_{i+1} = \phi_{X_i} \end{array}$$

si ottiene $s \vdash_\Delta True$, perciò quel percorso ha successo.

Utilizzando queste considerazioni si può costruire il Tableau risparmiandoci la generazione di tutto percorso del punto fisso;

Tableau & μ – *calcolo*

Costruzione del tableau

- La costruzione del tableau per analizzare le formule del μ – *calcolo* avviene utilizzando le stesse formule viste per l'*HML*; in più si devono aggiungere le regole di inferenza e le condizioni di successo per gli operatori di punto fisso;
- In particolare, per quanto riguarda gli operatori di punto fisso, si può terminare un percorso di ricerca quando si ottengono due nodi in cui la coppia *stesso stato - stessa formula* viene ripetuta, perchè questo vorrebbe dire che abbiamo incontrato un punto fisso;
- Naturalmente, dobbiamo preservare la portata dei punti fissi nidificati; per fare questo, ogni volta che un punto fisso viene incontrato nella ricerca, si introduce un'**etichetta univoca** $\mathcal{U}$, per denotare il punto fisso incontrato;
- L'etichetta e la formula denotata da essa vengono salvate poi in un **ambiente** Δ: l'ambiente si può definire in modo informale come una piccola memoria. Formalmente l'ambiente è una funzione che data una etichetta $\mathcal{U}$ restituisce la formula contenuta in essa;

Tableau & μ — *calcolo*

Regole di inferenza e soddisfacibilità

Le regole di inferenza per gestire gli operatori di punto fisso sono (queste regole sono scritte per μ ma valgono anche per ν):

$$1)\ \frac{s \vdash_{\Delta} \mu X.\phi_X}{s \vdash_{\Delta'} \mathcal{U}} \text{ dove } \Delta' = \Delta + [\mathcal{U} \mapsto \mu X.\phi_X]$$

$$2)\ \frac{s \vdash_{\Delta} \mathcal{U}}{s \vdash_{\Delta} \phi_{\mathcal{U}}} \text{ dove } \Delta(\mathcal{U}) = \mu X.\phi_X$$

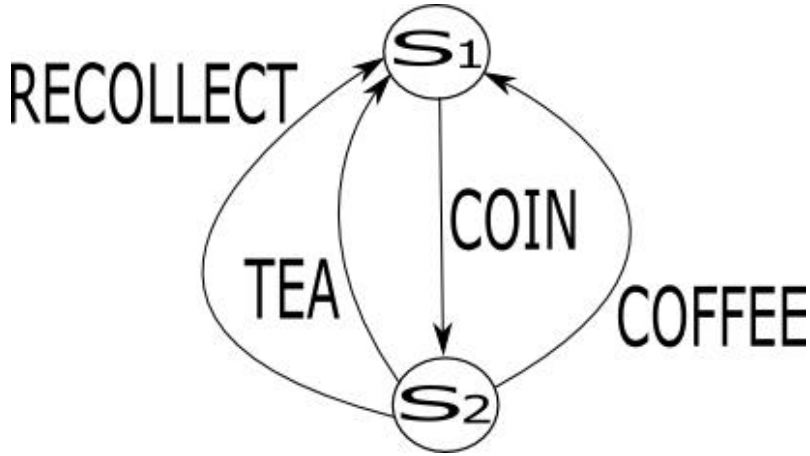
- **Terminazione**: La regola 2) può essere applicata solo quando $s \vdash_{\Delta} \mathcal{U}$ non è già apparsa in un "nodo antenato" $\Rightarrow$ Altrimenti, coppia di nodi con stesso stato - stessa formula $\Rightarrow$ Si è individuato un punto fisso $\Rightarrow$ $s \vdash_{\Delta} \mathcal{U}$ diventa una foglia;

- **Soddisfacibilità**: Una foglia etichettata con $s \vdash_{\Delta} \mathcal{U}$ ha successo sse $\Delta(\mathcal{U}) = \nu X.\phi_X$ (massimo punto fisso).

Tableau & μ — *calcolo*

Esempio - Macchinetta del caffè

Abbiamo il modello di una semplice macchinetta del caffè:



Verifichiamo se lo stato s_1 soddisfa la seguente formula:

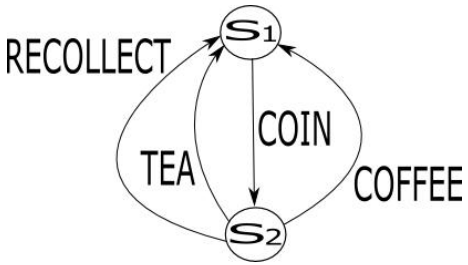
$$\nu X.([COIN] < RECOLLECT > True \wedge [Act]X)$$

La formula esprime la seguente proprietà: "È sempre vero che dopo aver inserito una moneta è possibile riprenderla".

Tableau & μ — *calcolo*

Esempio - Macchinetta del caffè

Il tableau che si può generare è:



$$s_1 \vdash_\emptyset \nu X.([COIN] < RECOLLECT > True \wedge [Act]X)$$
$$||$$
$$s_1 \vdash_\emptyset \nu X.\phi_X$$
$$\downarrow$$
$$s_1 \vdash_\Delta \mathcal{U}$$
$$\downarrow$$
$$s_1 \vdash_\Delta [COIN] < RECOLLECT > True \wedge [Act]\mathcal{U}$$
$$\downarrow$$
$$s_2 \vdash_\Delta < RECOLLECT > True \wedge [Act]\mathcal{U}$$

$$\swarrow \qquad \searrow$$

$$s_2 \vdash_\Delta < RECOLLECT > True \qquad s_2 \vdash_\Delta [Act]\mathcal{U}$$
$$\downarrow \qquad\qquad \downarrow$$
$$s_2 \vdash_\Delta True \qquad s_1 \vdash_\Delta \mathcal{U}$$

dove:
$\phi_X = [COIN] < RECOLLECT > True \wedge [Act]X$
$\Delta = \emptyset + [\mathcal{U} \mapsto \nu X.\phi_X]$

La prima foglia ha banalmente successo; Anche la seconda foglia ($s_1 \vdash_\Delta \mathcal{U}$) ha successo, perchè $\Delta(\mathcal{U}) = \nu X.\phi_X$, perciò si conclude che lo stato s_1 soddisfa la formula data.

Tableau & μ — *calcolo*

Esempio - Punti fissi annidati

I metodi basati su Tableau gestiscono anche punti fissi annidati. Ad esempio, prendiamo questo piccolo sistema di transizioni e proviamo che lo stato s soddisfa la seguente formula:

$$\nu X.(\mu Y. < a > True \vee < b > Y) \wedge [b]X$$

che dice: "Una transizione a è limitatamente raggiungibile lungo il percorso di transizioni b". Il sistema può essere rappresentato nel modo seguente:

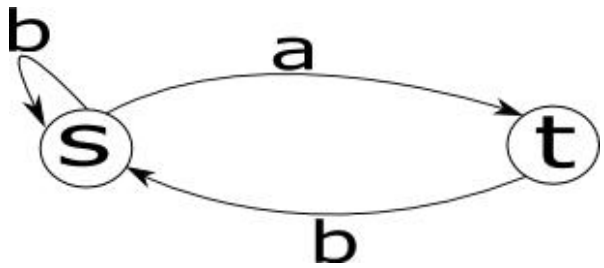
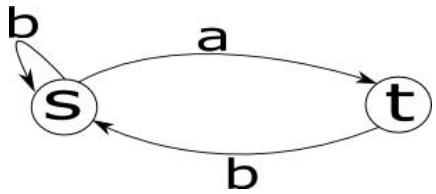


Tableau & μ — *calcolo*

Esempio - Punti fissi annidati

Il tableau che si può generare dalla formula data è il seguente:

$$
\begin{array}{c}
s \vdash_{\emptyset} \nu X.\phi_X \ \wedge [b]X \\
\downarrow \\
s \vdash_{\Delta_1} \mathcal{U}_1 \\
\downarrow \\
s \vdash_{\Delta_1} \phi_{\mathcal{U}_1} \ \wedge [b]\mathcal{U}_1 \\
\swarrow \qquad\qquad \searrow \\
s \vdash_{\Delta_1} \phi_{\mathcal{U}_1} \qquad\qquad s \vdash_{\Delta_1} [b]\mathcal{U}_1 \\
\downarrow \qquad\qquad\qquad\qquad \downarrow \\
s \vdash_{\Delta_2} \mathcal{U}_2 \qquad\qquad s \vdash_{\Delta_1} \mathcal{U}_1 \\
\downarrow \\
s \vdash_{\Delta_2} < a > True \vee < b > \mathcal{U}_2 \\
\downarrow \\
s \vdash_{\Delta_2} < a > True \\
\downarrow \\
t \vdash_{\Delta_2} True
\end{array}
$$



dove:
$\phi_X = \mu Y. < a > True \wedge < b > Y$
$\Delta_1 = \emptyset + [\mathcal{U}_1 \mapsto \nu X.\phi_X \ \wedge [b]X]$
$\Delta_2 = \Delta_1 + [\mathcal{U}_2 \mapsto \phi_{\mathcal{U}_1}]$

Tutte le foglie del Tableau rispettano le condizioni di soddisfacibilità perciò si conclude che lo stato s_1 rispetta la formula data.

Conclusioni

- Il model checking basato su Tableau è applicabile sia alle *logiche lineari*, sia alle *logiche branching*;
- La costruzione dell'albero generato dalla prova avviene in modo incrementale $\Rightarrow$ Solo i nodi strettamente necessari alla prova vengono generati $\Rightarrow$ Risparmio di spazio in memoria;
- Comportamenti infiniti (punti fissi) vengono gestiti in "modo finito" (cfr. Terminazione prova del $\mu - calcolo$), almeno per modelli a spazio degli stati finito;
- Algoritmi "ottimizzati" si possono fermare appena trovano una foglia che non ha successo, senza generare tutto il resto del Tableau $\Rightarrow$ Risparmio di tempo nell'analisi della formula;

Bibliografia

 B. Steffen, D. Schmidt. *Model checking, a tutorial introduction*.

 J. P. Katoen. *Concepts, algorithms, and tools for model checking*.

Fine