Three partition refinement algorithms

L'articolo presenta algoritmi risolutivi per tre problemi differenti.

L'algoritmo presentato migliora quello presentato da Hopcroft, pur utilizzando una strategia simile (ma migliorata).

$split(S, Q)$

Sia $S \subset U$ con U insieme finito, sia R una relazione binaria su U (cioè $R \subset U \times U$), e sia Q una partizione di U.

$$split(S, Q) = \begin{cases} B & \text{se B è un blocco di Q stabile rispetto a S (rispetto a R)} \\ \{B \cap R^{-1}(S), B - R^{-1}(S)\} & \text{altrimenti} \end{cases}$$

Diremo che S è uno *splitter* di Q se $split(S, Q) \boxed{?} Q$.

Proprietà di *split*

1. S è uno *splitter* per $Q \iff split(S, Q) \boxed{?} Q \iff Q$ è instabile rispetto a S
2. Se Q' è una rifinitura di Q, e Q è stabile rispetto a $S \implies Q'$ è stabile rispetto a S
3. Se Q è stabile rispetto a $S_1, S_2 \implies Q$ è stabile rispetto a $S_1 \cup S_2$
4. Monotonia: se Q' è una rifinitura di $Q \implies split(S, Q')$ è una rifinitura di $split(S, Q)$
5. Commutatività: $split(S, split(Q, P)) = split(Q, split(S, P))$

Algoritmo "naive"

Finchè è possibile trovare un insieme S tale che $S = \alpha \cup \beta$ con α, β blocchi di Q, e tale che S è uno *splitter* di Q, sosituisci Q con $split(S, Q)$.

Non è necessario usare come *splitter* unioni di blocchi della partizione attuale, ma questo consente di sviluppare l'algoritmo più veloce.

Algoritmo "fast"

Innanzitutto si preprocessa la partizione iniziale P dell'insieme U, sostituendo ogni blocco B con

$$B' = B \cap E^{-1}(U), \qquad B'' = B - E^{-1}(U)$$

E' evidente che tutti i blocchi B'' contengono tutti gli elementi $x : \nexists y \in U : E(x, y)$. Di conseguenza questi blocchi non verranno mai toccati da $split$, in quanto sono già stabili rispetto a qualsiasi sottoinsieme di U: prendendo un $S \subset U$ si ha che $B'' \cap E^{-1}(S) = \emptyset$ per qualsiasi B''.
Allora l'insieme dei B' è una partizione di $E^{-1}(U)$, e possiamo usare l'algoritmo soltanto su questi blocchi. Non possiamo unire i blocchi B'' perchè violeremmo la condizione che la partizione risultante deve essere una rifinitura della partizione iniziale.

Il miglioramente consiste nel mantenere due partizioni X, Q di U. Q è sempre una rifinitura di X, e Q è stabile rispetto ad ogni blocco di X. Inizialmente $Q = P, X = U$ (cioè X contiene un unico blocco). Si ripete il seguente algoritmo finchè non si ottiene $Q = X$

1. Trova un blocco $S \in X : S \notin Q$
2. Trova un blocco $B \in Q : B \subset S \wedge |B| \leq |S|/2$
3. Rimpiazza S in X con $B, S - B$
4. Rimpiazza B in Q con $split(S - B, split(B, Q))$

Questa modifica consente di scegliere gli *splitter* in modo più intelligenti. Nel caso peggiore (senza le due partizioni) anche con il preprocessamento l'algoritmo con *splitter* "casuali" ha complessità $O(mn)$.

Caso funzionale (da Hopcroft)

Suppongo che

$$\forall x \in U |E(\{x\})| = 1, \text{ cioè } \forall x \ \exists! y \in U : E(x, y) \qquad (1)$$

Sia Q una partizione di U. Sia $S = \cup_{i=1}^{n} b_i$ con $b_i \in Q$. Suppongo Q stabile rispetto a S. Sia $B \subset S$.
Allora

$$split(B, Q) \text{ è stabile rispetto a } S - B$$

Infatti, sia $B_1 \in Q$.

- se B_1 era un blocco di Q già stabile rispetto a B (quindi B_1 non è cambiato), allora
  - $B_1 \subset E^{-1}(B) \implies B_1 \cap E^{-1}(S - B) = \emptyset$, perchè $B \cap S - B = \emptyset$ e per l'ipotesi (1)
  - $B_1 \cap E^{-1}(B) = \emptyset$. Ricordando che B_1 era già un blocco di Q stabile rispetto a S
    - Non può essere $B_1 \subset E^{-1}(S)$, perchè $B_1 \cap E^{-1}(B) = \emptyset$ e $B \subset S$
    - $B_1 \cap E^{-1}(S) = \emptyset \implies B_1 \cap E^{-1}(S - B) = \emptyset$
- se B_1 è stato generato da uno *split*
  - $B_1 = \widetilde{B} \cap E^{-1}(B)$ per qualche blocco $\widetilde{B} \in Q \implies$ deve essere chiaramente $B_1 \cap E^{-1}(S - B) = \emptyset$ per la (1)
  - $B_1 = \widetilde{B} - E^{-1}(B)$ per qualche blocco $\widetilde{B} \in Q$
    - $\widetilde{B} \subset E^{-1}(S) \implies \widetilde{B} - E^{-1}(B) \subset E^{-1}(S - B)$
    - $\widetilde{B} \cap E^{-1}(S) = \emptyset \implies \widetilde{B} \cap E^{-1}(B) = \emptyset \implies B_1 = \widetilde{B} - E^{-1}(B) = \widetilde{B}$, ma questo caso è già stato trattato.