Raffinement automatique de partition

Raffinement de partitions

- Objectif: trouver une partition $\simeq$ stable adéquate automatiquement.
- Motivation: réduire la taille d'un système avant de vérifier.
  - exemple: vérification de plusieurs propriétés d'un même système.
  - exemple: réduire la taille de P_1 et de P_2 avant de considérer $P = P_1 \| P_2$.

CPO des partitions

- Définition: $\simeq_1$ raffine $\simeq_2$ si $a \simeq_1 b$ implique $a \simeq_2 b$
  autrement dit, $\simeq_1 \subset \simeq_2$ comme relations d'équivalence.
  Cela implique que $\simeq_1$ a plus de régions que $\simeq_2$.
- Définition: $join(\simeq_1, \simeq_2)$ est la clôture transitive de l'union des deux relations d'équivalence.
  (autrement dit, s et t sont dans la même classe s'il existe $u_1, u_2, \ldots u_n$ tels que $s \simeq_1 u_1 \simeq_2 u_2 \simeq_1 u_3 \ldots u_n \simeq t$).

- Propriétés: si $\simeq_1$ et $\simeq_2$ sont deux partitions, alors
  1. $join(\simeq_1, \simeq_2)$ est une partition.
  2. $\simeq_1$ et $\simeq_2$ raffinent $join(\simeq_1, \simeq_2)$.

Démonstration: Exercice

- Théorème: le $join$ de deux relations stables est une relation stable.

Démonstration: Exercice

- $join$ et les résultats précédents s'étendent facilement aux ensembles de partitions: $join(\simeq_1, \ldots, \simeq_n)$ est défini comme la clôture transitive de l'union des relations d'équivalence $\simeq_1,\ldots,\simeq_n$.

  Démonstration: exercice.

- Proposition: étant donnée $\simeq$,
  1. l'ensemble de toutes les partitions qui sont stables et qui raffinent $\simeq$ possède un élément minimal (en nombre de classes) (c'est un ordre partiel complet);
  2. cet élément, noté $min_M(\simeq)$, est le $join$ de toutes les partitions stables qui raffinent $\simeq$.

Démonstration. Il existe au moins une partition stable qui raffine $\simeq$: l'identité. Le $join$ de toutes les partitions stables qui raffinent $\simeq$ est une partition stable. Soit $\simeq'$ une partition stable qui raffine $\simeq$. Le $join$ de toutes les partitions stables qui raffinent $\simeq$ contient en particulier $\simeq'$, et donc possède moins de classes que $\simeq'$.

- Remarques:
  - si $\simeq = id$ alors $min_M(\simeq) = id$.
  - si $\simeq$ est "tous les états équivalents" alors $min_M(\simeq)$ aussi.
  - si $\simeq$ est stable, alors $min_M(\simeq) = \simeq$

Problème du raffinement

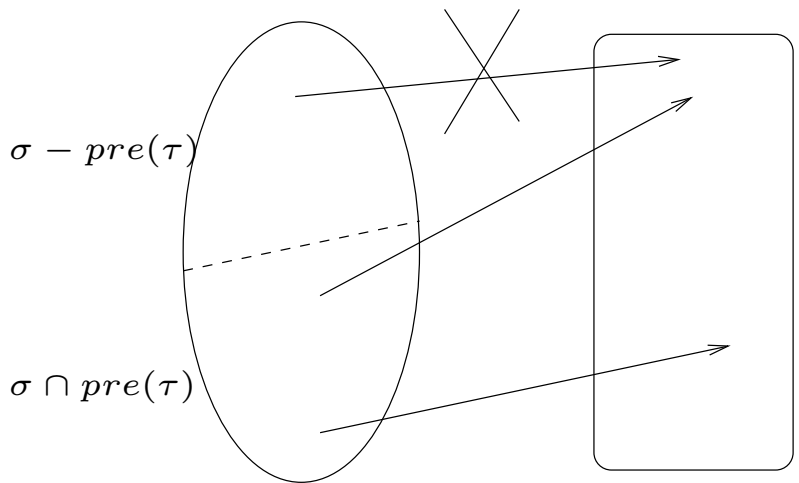
- Problème du raffinement:
  - Données:
    - M système de transition,
    - $\simeq^I$ partition initiale.
  - Réponse: $min_M(\simeq^I)$.
- Application: Pour résoudre un problème d'invariance $M \models^? \forall\Box p$, on peut
  1. considérer la partition initiale $\simeq^I = \{[\![p]\!], [\![\neg p]\!]\}$.
  2. calculer ensuite $\simeq = min_M(\simeq^I)$.
  3. resoudre le problème $Atteignable(M/\simeq, [[p]]/\simeq)$ qui concerne un système $M/\simeq$ avec moins d'états.

Calcul automatique de $min_M(\simeq)$

Caractérisation des régions stables

- Définition: $pre(\tau)$ dénote les prédécesseurs d'une région τ: $pre(\tau) = \{s | \exists t \in \tau, R(s,t)\}$.
- Une région σ est dite stable par rapport à une région τ si $\sigma \subset pre(\tau)$ ou $\sigma \cap pre(\tau) = \emptyset$.

  (Autrement dit: il existe $s \in \sigma$ avec un $t \in \tau$ tel que $R(s,t)$, implique pour tous les $s \in \sigma$, il y a un $t \in \tau$ avec $R(s,t)$.)



- Une partition $\simeq$ est dite stable par rapport à région τ si toutes ses classes d'équivalence le sont.
- Proposition: Une partition $\simeq$ est stable ssi toutes ses régions sont stables par rapport à toutes ses régions.

Démonstration. Supposons $\simeq$ stable. Soient σ, τ deux régions. Supposons qu'il existe $s \in \sigma$ avec un $t \in \tau$ tel que $R(s,t)$. Pour tout $s' \in \sigma$, puisque $s \simeq s'$, par stabilité, il existe t' avec $t \simeq t'$ et $R(s',t')$, i.e. il y a un $t' \in \tau$ avec $R(s',t')$.
Réciproquement, soient s, t, s' avec $R(s,t)$ et $s \simeq s'$. Considérons σ la classe d'équivalence de s, et τ la classe d'équivalence de t. On a $s, s' \in \sigma$. Puisque σ et stable par rapport à τ, pour tout $s'' \in \sigma$ il y a un $t'' \in \tau$, (donc avec $t'' \simeq t$) tel que $R(s'',t'')$. En particulier pour $s'' = s'$ on obtient $t' = t''$ tel que $R(s',t')$ et $t \simeq t'$.

- Stratégie pour calculer $min_M(\simeq)$: essayer de stabiliser la partition vis à vis de toutes ses régions.

Principe

- Stratégie de base:
  1. si $\simeq$ est non stable, alors il existe deux régions σ, τ avec σ non-stable par rapport à τ.
  2. découper σ en deux régions suivant la région prédécesseur de τ.
- Découpage d'une région σ par un sous-ensemble τ': $Split(\sigma, \tau')$ =
  1. $\{\sigma\}$ si $\sigma \subset \tau'$ ou $\sigma \cap \tau' = \emptyset$
  2. $\{\sigma \cap \tau', \sigma - \tau'\}$ sinon
- Découpage d'une partition $\simeq$ par un sous-ensemble τ': $Split(\simeq, \tau')$ = nouvelle partition obtenue en découpant chaque région de $\simeq$ par τ'.

Propriétés

- Stabilisation d'une partition $\simeq$ par rapport à une région τ: $Stabilise(\simeq, \tau) = Split(\simeq, pre(\tau))$.

- Propriété: si τ est un bloc de $\simeq$, alors $min_M(\simeq)$ raffine $Stabilise(\simeq, \tau)$.
  (I.e. la stabilisation par rapport à un bloc ne cause jamais de découpage inutile.)

Démonstration: supposons $s\ min_M(\simeq)\ s'$. Puisque $min_M(\simeq)$ raffine $\simeq$, on a $s \simeq s'$. Supposons par l'absurde que $\neg(s\ Split(\simeq, pre(\tau))\ s')$. Quitte à inverser le rôle de s et s', on peut supposer $s \in pre(\tau)$ et $s' \notin pre(\tau)$. Soit $t \in \tau$ tel que $R(s,t)$. La stabilité de $min_M(\simeq)$ implique qu'il doit exister un t' avec $R(s',t')$ et $t\ min_M(\simeq)\ t'$. $t\ min_M(\simeq)\ t'$ implique $t \simeq t'$, puisque $min_M(\simeq)$ raffine $\simeq$, et donc ce $t' \in \tau$ avec $R(s',t')$ est en contradiction avec $s' \notin pre(\tau)$.

- Propriété: toute partition qui raffine $Stabilise(\simeq, \tau)$ est stable par rapport à τ.

Démonstration. Une région σ' de cette partition est incluse dans une région de $Stabilise(\simeq, \tau)$. Si cette région est une région σ de $\simeq$ inchangé par l'opération $Split(\sigma, pre(\tau))$, alors on a soit $\sigma' \subset \sigma \subset pre(\tau)$ ou $\sigma' \cap pre(\tau) \subset \sigma \cap pre(\tau) = \emptyset$. Si cette région correspond à $\sigma \cap pre(\tau)$ pour une région σ de $\simeq$, alors on a $\sigma' \subset \sigma \cap pre(\tau) \subset pre(\tau)$. Si cette région correspond à $\sigma - pre(\tau)$ pour une région σ de $\simeq$, alors on a $\sigma' \subset \sigma - pre(\tau)$ et donc $\sigma' \cap pre(\tau) = \emptyset$.

Algorithme schématique

- Données: un système de transitions M et une partition initiale $\simeq^I$.
- Réponse: $min(\simeq^I)$.
- Variables:
  $\simeq$: M-partition (= ensemble de régions particulier),
  $done$: ensemble de régions.
- Algorithme:
  1. $\simeq := \simeq^I$; $done := \{S\}$
  2. tant que $\simeq \not\subset done$ faire
     - choisir une région τ de $\simeq$ telle que $\tau \notin done$
     - $\simeq := Stabilise(\simeq, \tau)$
     - $done := done \cup \{\tau\}$
  3. retourner($\simeq$).

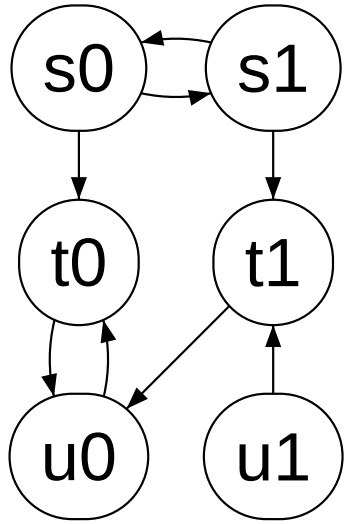
- Propositions (invariants de l'algorithme)
  1. $min_M(\simeq^I)$ est un raffinement de $\simeq$ à toute étape.
  2. Chaque ensemble dans $done$ est un bloc de $\simeq$.
  3. La partition $\simeq$ est stable vis à vis de toutes les élements de $done$.

  Démonstration: ces invariants sont des conséquences immédiates par récurrence des deux propriétés précédentes.

- Complexité: $O(2^n)$ itérations, où n est le nombre de régions de $min_M(\simeq^I)$.
  Démonstration: si $min_M(\simeq^I)$ possède n-classes d'équivalence, alors il y au pire 2^n blocs. Chaque itération de la boucle tant que ajoute au plus un bloc de $min_M(\simeq^I)$ a l'ensemble $done$. La boucle tant que est exécutée au plus 2^n fois.

Exemple

- Soit le système M:



- Calculer $min_M(\simeq^I)$ avec $\simeq^I = \{\{u_0\}, S - \{u_0\}\}$.

- Réponse:
  1. Etat initial: $Done = \{S\}$, $\simeq = \{\{u_0\}, S - \{u_0\}\}$.
  2. Après itération 1 avec choix $\tau = \{u_0\}$: $pre(\tau) = \{t_0, t_1\}$,
     $Done = \{S, \{u_0\}\}$, $\simeq = \{\{u_0\}, \{t_0, t_1\}, \{s_0, s_1, u_1\}\}$.
  3. Après itération 2 avec $\tau = \{s_0, s_1, u_1\}$: $pre(\tau) = \{s_0, s_1\}$,
     $Done = \{S, \{u_0\}, \{s_0, s_1, u_1\}\}$, $\simeq = \{\{u_0\}, \{t_0, t_1\}, \{s_0, s_1\}, \{u_1\}\}$.
  4. Après itération 3 avec $\tau = \{t_0, t_1\}$: $pre(\tau) = \{s_0, s_1, u_0, u_1\}$,
     $Done = \{S, \{u_0\}, \{s_0, s_1, u_1\}, \{t_0, t_1\}\}$,
     $\simeq = \{\{u_0\}, \{t_0, t_1\}, \{s_0, s_1\}, \{u_1\}\}$.
  5. Après itération 4 avec $\tau = \{u_1\}$: $pre(\tau) = \emptyset$,
     $Done = \{S, \{u_0\}, \{s_0, s_1, u_1\}, \{t_0, t_1\}, \{u_1\}\}$,
     $\simeq = \{\{u_0\}, \{t_0, t_1\}, \{s_0, s_1\}, \{u_1\}\}$.
  6. Après itération 5 avec $\tau = \{s_0, s_1\}$: $pre(\tau) = \{s_0, s_1\}$,
     $Done = \{S, \{u_0\}, \{s_0, s_1, u_1\}, \{t_0, t_1\}, \{u_1\}, \{s_0, s_1\}\}$,
     $\simeq = \{\{u_0\}, \{t_0, t_1\}, \{s_0, s_1\}, \{u_1\}\}$.
  7. Réponse: $\simeq = \{\{u_0\}, \{t_0, t_1\}, \{s_0, s_1\}, \{u_1\}\}$.

Amélioration: algorithme quadratiqu

- Principe: garder la partition précédente et stabiliser par rapport aux classes d'équivalence de $\simeq^{prev}$.
- Données: un système de Transition M et une partition initiale $\simeq^{I}$.
- Réponse: $min(\simeq^{I})$.
- Variables: $\simeq$,$\simeq^{prev}$ M-partitions
- Algorithme:

1. $\simeq := \simeq^{I}$; $\simeq^{prev} := \{S\}$
2. tant que $\simeq \neq \simeq^{prev}$ faire
   $\simeq^{prev} := \simeq$
   pour tout $\tau \in \simeq^{prev}$
   $\simeq := Stabilise(\simeq, \tau)$

retourner($\simeq$).

- Cet algorithme est correct car c'est une instance particulière de l'algorithme précédent (rappel: stabiliser par une région stable ne fait rien).
- Proposition: si $min_M(\simeq^I)$ possède n-classes d'équivalence alors la boucle tant que est exécutée au plus n-fois.

  Démonstration: chaque itération fait croître le nombre de classes de $\simeq$.

  On peut faire mieux: Paige-Tarjan on proposé un algorithme subquadratique.
- Implémentation: peut s'implémenter avec des BDDs.

Raffinement de partitions atteignables

Raffinement de partition atteignable

- Motivation: On s'intéresse seulement à la portion du graphe quotient atteignable.
- Définition: $min^R(\simeq)$ dénote les classes d'équivalence atteignables de $min(\simeq)$.
- Définition: $M/min^R(\simeq)$ est le quotient minimal atteignable de M.
- Proposition: soit σ^T un bloc d'une partition $\simeq$. σ^T est atteignable dans M ssi $\sigma^T \cap \tau$ l'est dans le quotient minimal atteignable pour une certaine région $\tau \in min^R(\simeq)$.

Raffinement de partition atteignable

- Données:
  - un système de transitions M,
  - une partition initiale $\simeq^I$.
- Réponse: $min^R(\simeq^I)$.
- Principe: on découpe seulement les régions qui sont connues pour être atteignables en maintenant l'atteignabilité entre régions.
- Définition: $post(\tau)$ dénote les successeurs d'une région τ: $post(\tau) = \{\sigma | \exists s \in \tau, t \in \sigma, R(s,t)\}$.

Algorithme

1. $\simeq := \simeq^I$; $\sigma^R := \emptyset$
2. répète

   commentaire: $min^I(\simeq)$ est un raffinement de $\simeq$

   $\simeq_R := \{\sigma \in \simeq \mid \sigma \cap \sigma^R \neq \emptyset\}$

   commentaire: σ^R contient seulement des états atteignables, et au plus un état par $\sigma \in \simeq$

   $F := \{\sigma \in \simeq - \simeq_R \mid \sigma \cap (S_0 \cup post(\sigma^R)) \neq \emptyset\}$

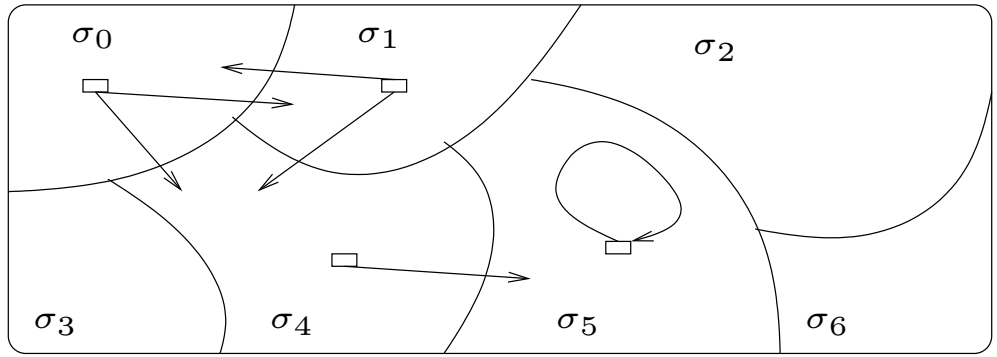
   $V := \{(\tau, v) \in \simeq_R \times \simeq \mid \tau$ n'est pas stable par rapport à $v\}$

Cherche:	ou Découpe:
Choisir $\sigma \in F$	Choisir $(\tau, v) \in V$
Choisir $s \in \sigma \cap (S_0 \cup post(\sigma^R))$	Stabiliser τ / v

   $\sigma^R := \sigma^R \cup \{s\}$

   jusqu'à $F = \emptyset$ ou $V = \emptyset$
3. retourner $\simeq_R$.

- L'algorithme peut s'implémenter avec des BDDs.
- Sa performance dépend de la stratégie choisie entre "cherche"/ "découpe". Algorithme de Lee-Yannakis: "cherche" prioritaire: stratégie ((Cherche)* Découpe)*).
- Illustration:



- L'algorithme ne retourne pas $\sigma_3, \sigma_2, \sigma_5$ car ces partitions ne sont pas atteignables.

Exemple d'exécution:

- initialement $\simeq = \{\sigma_0, S - \sigma_0\}$
- $S - \sigma_0$ est découpé en $\{\sigma_1, S - \sigma_0 - \sigma_1\}$ car $S - \sigma_0$ est non stable par rapport à σ_0.
- σ_1 est détecté atteignable.
- $S - \sigma_0 - \sigma_1$ est découpé en $\{\sigma_4, S - \sigma_0 - \sigma_1 - \sigma_4\}$ car non stable par rapport à σ_1.
- σ_4 est détecté atteignable.
- $S - \sigma_0 - \sigma_1 - \sigma_4$ est découpé en $\{\sigma_5, S - \sigma_0 - \sigma_1 - \sigma_4 - \sigma_5\}$ car est non stable par rapport à σ_4.
- σ_5 est détecté atteignable.
- on retourne finalement $\{\sigma_0, \sigma_1, \sigma_4, \sigma_5\}$.