Cinquième partie

TLA^+– la logique

Objectifs

- Exprimer des propriétés vérifiées par une spécification TLA$^+$
- Exprimer l'équité garantissant la progression
- Démontrer des liens entre spécifications (équivalence, raffinage)
- Pouvoir vérifier ces propriétés de manière mécanisée (automatiquement – vérification de modèles –, ou manuellement – preuve axiomatique –)

Plan

1. LTL et TLA$^+$
   - Logique TLA$^+$
   - Raffinage
2. Preuve axiomatique
3. Vérification de modèles

La logique TLA$^+$

Expressions logiques

Expressions de LTL avec $\Box$, $\Diamond$, $\leadsto$ (leads-to) et variables primées + quantificateurs $\forall, \exists$.

Pas de $\mathcal{U}$, ni de $\mathcal{W}$, mais :

$$\Box(p \Rightarrow (p\mathcal{W}q)) \quad = \quad \Box(p \Rightarrow (p' \vee q))$$
$$\Box(p \Rightarrow (p\mathcal{U}q)) \quad = \quad \Box(p \Rightarrow (p' \vee q)) \wedge \Box(p \Rightarrow \Diamond q)$$

Équité / *Fairness*

ENABLED

ENABLED $\mathcal{A}$ est la fonction d'état qui est vraie dans l'état s ssi il existe un état t accessible depuis s par l'action $\mathcal{A}$.

Weak/Strong Fairness

- $\mathrm{WF}_e(\mathcal{A}) \triangleq \Box\Diamond\neg(\textsc{enabled}\ \langle\mathcal{A}\rangle_e) \vee \Box\Diamond\langle\mathcal{A}\rangle_e$
  si $\mathcal{A}$ est constamment déclenchable, elle sera déclenchée.
- $\mathrm{SF}_e(\mathcal{A}) \triangleq \Diamond\Box\neg(\textsc{enabled}\ \langle\mathcal{A}\rangle_e) \vee \Box\Diamond\langle\mathcal{A}\rangle_e$
  si $\mathcal{A}$ est infiniment souvent déclenchable, elle sera déclenchée.

Forme d'une spécification TLA$^+$

En général, une spécification TLA$^+$ est une conjonction

$$\mathcal{I} \land \Box[\mathcal{N}]_v \land \mathcal{E}$$

- $\mathcal{I}$ = prédicat d'état décrivant les états initiaux
- $\mathcal{N}$ = disjonction d'actions $\mathcal{A}_1 \lor \mathcal{A}_2 \lor \mathcal{A}_3 \lor \ldots$
- $\mathcal{E}$ = conjonction de contraintes d'équité portant sur les actions : $\mathrm{WF}_v(A_1) \land \mathrm{SF}_v(A_3) \land \ldots$

Raffinage de spécification

Raffinage simple

Une spécification (concrète) Pc raffine une spécification (abstraite) Pa si $Pc \Rightarrow Pa$: tout ce que fait Pc est possible dans Pa.

Cela signifie que si $Pa \models P$ pour une propriété LTL quelconque, alors $Pc \models P$.

Raffinage – exemple

Somme abstraite

MODULE *somme*1

EXTENDS *Naturals*

CONSTANT N

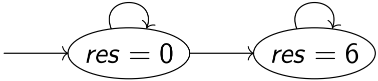
VARIABLE *res*

$$TypeInvariant \triangleq res \in Nat$$

$$Init \triangleq res = 0$$
$$Next \triangleq res' = ((N+1) * N) \div 2$$
$$Spec \triangleq Init \land \Box[Next]_{res} \land \mathrm{WF}_{res}(Next)$$

Raffinage – exemple

Graphe des exécutions pour $N = 3$



Raffinage – exemple

Somme plus concrète

MODULE *somme2*

EXTENDS *Naturals*

CONSTANT *N*

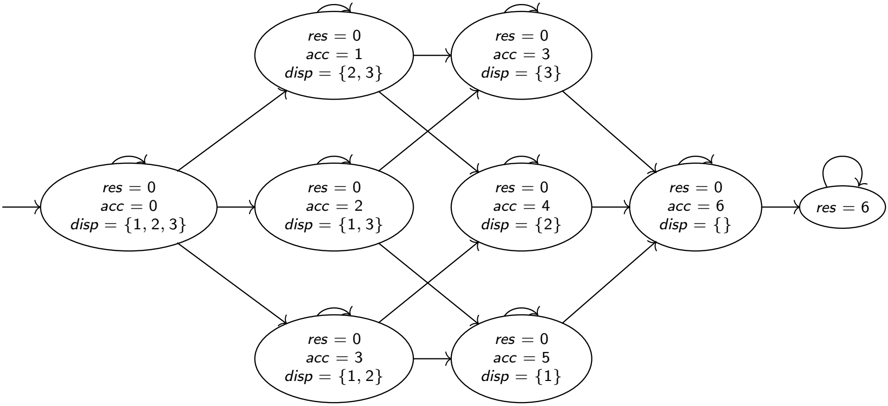
VARIABLE *res*, *acc*, *disp*

$$TypeInvariant \triangleq res \in Nat \land acc \in Nat \land disp \in \text{SUBSET } 1 \,..\, N$$

$$\begin{aligned}
Init &\triangleq res = 0 \land acc = 0 \land disp = 1 \,..\, N \\
Next &\triangleq \lor \exists i \in disp : acc' = acc + i \land disp' = disp \setminus \{i\} \\
&\qquad\qquad\qquad\qquad\qquad \land \text{UNCHANGED } res \\
&\qquad \lor disp = \{\} \land res' = acc \land \text{UNCHANGED } \langle disp, acc \rangle \\
Spec &\triangleq Init \land \Box[Next]_{\langle res, disp, acc \rangle} \land \mathrm{WF}_{\langle res, disp, acc \rangle}(Next)
\end{aligned}$$

Raffinage – exemple

Graphe des exécutions pour $N = 3$



Décomposition : introduction de transitions intermédiaires.

Raffinage – exemple

Somme2 raffine somme1

MODULE *somme2_raffine_somme1*

$\text{EXTENDS } somme2$

$Orig \triangleq \text{INSTANCE } somme1$

$Raffinement \triangleq Orig!Spec$

$\text{THEOREM } Spec \Rightarrow Orig!Spec$

Raffinage – exemple

Somme concrète

MODULE *somme3*

EXTENDS *Naturals*

CONSTANT *N*

VARIABLE *res*, *acc*, *i*

$$TypeInvariant \triangleq res \in Nat \land acc \in Nat \land i \in 1 \,..\, N$$

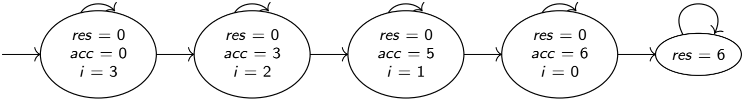
$$Init \triangleq res = 0 \land acc = 0 \land i = N$$

$$\begin{aligned} Next \triangleq\ & \lor\ i > 0 \land acc' = acc + i \land i' = i - 1 \land \text{UNCHANGED } res \\ & \lor\ i = 0 \land res' = acc \land \text{UNCHANGED } \langle i, acc \rangle \end{aligned}$$

$$Spec \triangleq Init \land \Box[Next]_{\langle res, i, acc \rangle} \land \mathrm{WF}_{\langle res, i, acc \rangle}(Next)$$

Raffinage – exemple

Graphe des exécutions pour N $= 3$



Réduction du non-déterminisme + changement de représentation (raffinement de données) $disp = 1..i$

Raffinage – exemple

Somme3 raffine somme2

MODULE *somme3_raffine_somme2*

$\text{EXTENDS } somme3$

$dispMapping \triangleq 1 \,..\, i$

$Orig \triangleq \text{INSTANCE } somme2 \text{ WITH } disp \leftarrow dispMapping$

$Raffinement \triangleq Orig!Spec$

$\text{THEOREM } Spec \Rightarrow Orig!Spec$

Plan

Règles de preuve – simple temporal logic

$$\frac{F \text{ prouvable en logique propositionnelle}}{\Box F}\,\mathrm{STL1} \qquad \frac{F \Rightarrow G}{\Box F \Rightarrow \Box G}\,\mathrm{STL4}$$

$$\frac{}{\Box F \Rightarrow F}\,\mathrm{STL2} \qquad \frac{}{\Box\Box F = \Box F}\,\mathrm{STL3}$$

$$\frac{}{\Box(F \wedge G) = (\Box F) \wedge (\Box G)}\,\mathrm{STL5} \qquad \frac{}{\Diamond\Box F \wedge \Diamond\Box G = \Diamond\Box(F \wedge G)}\,\mathrm{STL6}$$

Règles de preuve – TLA$^+$ invariant

$$\frac{P \wedge (v' = v) \Rightarrow P'}{\Box P = (P \wedge \Box[P \Rightarrow P']_v)}\text{TLA1} \quad \Bigg| \quad \frac{P \wedge [\mathcal{A}]_{v_1} \Rightarrow Q \wedge [\mathcal{B}]_{v_2}}{\Box P \wedge \Box[\mathcal{A}]_{v_1} \Rightarrow \Box Q \wedge \Box[\mathcal{B}]_{v_2}}\text{TLA2}$$

TLA1 : principe d'induction pour prouver $\Box P$ à partir de l'état initial et de la conservation de P. TLA2 : le raffinage de spécifications se ramène au raffinage des actions.

$$\frac{I \wedge [\mathcal{N}]_v \Rightarrow I'}{I \wedge \Box[\mathcal{N}]_v \Rightarrow \Box I}\text{INV1} \quad \Bigg| \quad \frac{}{\Box I \Rightarrow (\Box[\mathcal{N}]_v = \Box[\mathcal{N} \wedge I \wedge I']_v)}\text{INV2}$$

INV1 : preuve par induction d'un invariant
Hypothèse : I est préservé par $\mathcal{N}$ et le bégaiement.
Conclusion : si I est initialement vrai et toute transition vérifie $\mathcal{N}$ ou bégaiement ($\Box[\mathcal{N}]$), alors I est un invariant ($\Box I$).
INV2 : injection d'un invariant dans la spécification.

Règles de preuve – TLA$^+$ vivacité avec équité faible

$$\frac{\begin{array}{c} P \wedge [\mathcal{N}]_v \Rightarrow (P' \vee Q') \\ P \wedge \langle \mathcal{N} \wedge \mathcal{A} \rangle_v \Rightarrow Q' \\ P \Rightarrow \text{ENABLED } \langle \mathcal{A} \rangle_v \end{array}}{\Box[\mathcal{N}]_v \wedge WF_v(\mathcal{A}) \Rightarrow (P \leadsto Q)} \text{WF1}$$

Hypothèses :

1. si on a P, en faisant une transition quelconque ($[\mathcal{N}]$), on conserve P ou on établit Q ;
2. Il y a une action $\mathcal{A}$ qui établit Q ;
3. Quand P est vrai, l'action $\mathcal{A}$ est faisable.

Alors, sous contrainte d'équité faible sur $\mathcal{A}$, si P est vrai, $\mathcal{A}$ doit finir par avoir lieu (car P reste constamment vrai au moins jusqu'à établir Q et P garantit que $\mathcal{A}$ est faisable), et donc Q finira par être vrai.

Règles de preuve – TLA$^+$ vivacité avec équité forte

$$\frac{\begin{array}{c} P \wedge [\mathcal{N}]_v \Rightarrow (P' \vee Q') \\ P \wedge \langle \mathcal{N} \wedge \mathcal{A} \rangle_v \Rightarrow Q' \\ \Box P \wedge \Box[\mathcal{N}]_v \wedge \Box F \Rightarrow \Diamond \text{ENABLED } \langle \mathcal{A} \rangle_v \end{array}}{\Box[\mathcal{N}]_v \wedge SF_v(\mathcal{A}) \wedge \Box F \Rightarrow (P \leadsto Q)} \text{SF1}$$

Hypothèses :

1. si on a P, en faisant une transition quelconque ($[\mathcal{N}]$), on conserve P ou on établit Q ;
2. Il y a une action $\mathcal{A}$ qui établit Q ;
3. Si P est constamment vrai, l'action $\mathcal{A}$ finira par être faisable.

Alors, sous contrainte d'équité forte sur $\mathcal{A}$, si P est vrai, $\mathcal{A}$ doit finir par avoir lieu (car P reste constamment vrai au moins jusqu'à établir Q et P garantit que $\mathcal{A}$ sera faisable, donc est infiniment souvent faisable), et donc Q finira par être vrai.

($\Box F$ est un invariant qui facilite en général la preuve du 3)

Règles de preuve dérivées

$$\frac{\Box(P \Rightarrow \Box P) \land \Diamond P}{\Diamond\Box P}\,\mathrm{LDSTBL}$$

$\Box(P \Rightarrow \Box P)$ signifie que P est stable : une fois vrai, il le reste. Combiné avec $\Diamond P$ (un jour P sera vrai), on obtient que P est finalement toujours vrai.

$$\frac{P \leadsto Q \land Q \leadsto R}{P \leadsto R}\,\mathrm{TRANS}$$

Transitivité du $\leadsto$.

Plan

1. LTL et TLA$^+$
   - Logique TLA$^+$
   - Raffinage
2. Preuve axiomatique
3. Vérification de modèles

Vérification de modèles

Principe

Construire le graphe des exécutions et étudier la propriété.

- $\Box P$, où P est un prédicat d'état (sans variable primée) : au fur et à mesure de la construction des états.
- $\Box P(v, v')$, où $P(v, v')$ est un prédicat de transition (prédicat non temporel avec variables primées et non primées) : au fur et à mesure du calcul des transitions.
- Vivacité $\Diamond P$, $P \leadsto Q\ldots$: une fois le graphe construit, chercher un cycle qui respecte les contraintes d'équité et qui invalide la propriété.

Uniquement sur des modèles finis, et, pratiquement, de petites tailles.

Complexité

Soit $|S|$ le nombre d'états d'un système $\mathcal{S} = \langle S, I, R \rangle$ et $|F|$ la taille (le nombre d'opérateurs temporels) d'une formule LTL F.
La complexité en temps (et espace) pour vérifier $\mathcal{S} \models F$ est $O(|S| \times 2^{|F|})$.

Vérificateur TLC

Le vérificateur de modèles TLC sait vérifier :

- les spécifications avec des actions gardées ;
- (efficacement) les invariants sans variables primées : $\Box P$ où P est un prédicat d'état ;
- les formules de sûreté pure avec variables primées et bégaiement : $\Box[P]_v$ où P est un prédicat de transition ;
- $P \leadsto Q$ où P et Q sont des prédicats d'état (*sans* variables primées) ;
- les formules combinant $\Box, \Diamond$ *sans* variables primées.

Note : l'espace d'états du système et des formules doit être fini : toute quantification bornée par exemple.

Conclusion

- Propriétés de sûreté et de vivacité : LTL (logique temporelle linéaire)
- Équité pour isoler les contraintes de progression
- Vérification mécanisée (par modèle ou par preuve axiomatique)