Du calcul des durées aux automates symboliques

Laure Danthony

Stage de DEA

Sous la direction de Nicolas Halbwachs
Verimag - Grenoble - France

mars-juin 2003

Plan

- Contexte / Objet de l'étude
- Une idée de la logique utilisée : QDDC
- Traduction de QDDC
- Deux fragments distincts
- Conclusion

Contexte de l'étude

- Vérification automatique de programmes réactifs

Contexte de l'étude

- ⋆ Vérification automatique de programmes réactifs
- ⋆ Model checking (abstractions)

Contexte de l'étude

- ⋆ Vérification automatique de programmes réactifs
- ⋆ Model checking (abstractions)
- ⋆ Analyse approchée (interprétation abstraite)

Contexte de l'étude

- ⋆ Vérification automatique de programmes réactifs
- ⋆ Model checking (abstractions)
- ⋆ Analyse approchée (interprétation abstraite)

➥ cadre symbolique + variables numériques (vérification conservative)

Le langage Lustre

- Modèle synchrone (E/S synchronisées)

Le langage Lustre

- ⋆ Modèle synchrone (E/S synchronisées)
- ⋆ Notion de flot de variables

Le langage Lustre

- ⋆ Modèle synchrone (E/S synchronisées)
- ⋆ Notion de flot de variables

- ⋆ Définitions par équation $s_1 = e_1 \wedge e_2 \wedge \mathtt{pre}(e_3)$
  $(\forall n, s_1^{(n)} = e_1^{(n)} \wedge e_2^{(n)} \wedge e_3^{(n-1)})$

Le langage Lustre

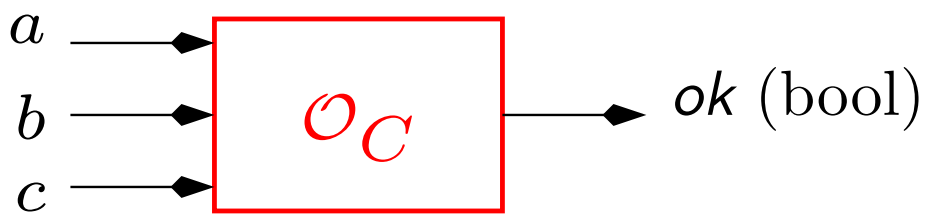
- ⋆ Modèle synchrone (E/S synchronisées)
- ⋆ Notion de flot de variables

- ⋆ Définitions par équation $s_1 = e_1 \wedge e_2 \wedge \mathtt{pre}(e_3)$
  $(\forall n, s_1^{(n)} = e_1^{(n)} \wedge e_2^{(n)} \wedge e_3^{(n-1)})$

➥ automates symboliques

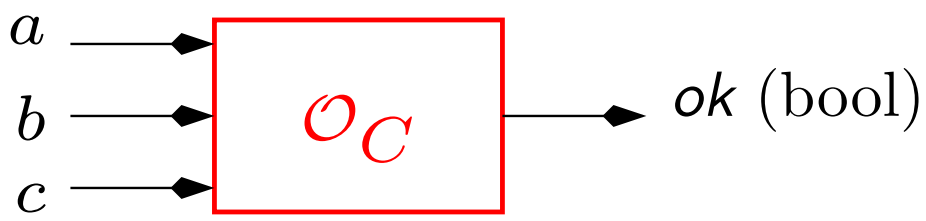
Observateurs

⋆ Propriété $\Rightarrow$ Automate Observateur



Observateurs

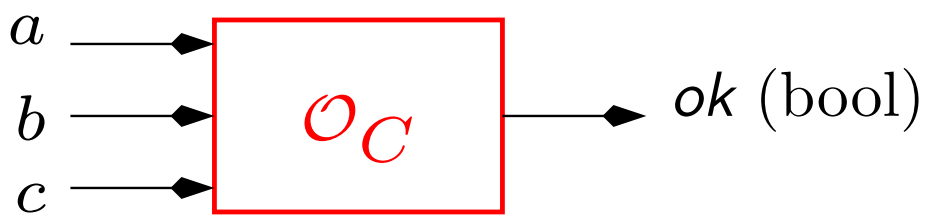
- Propriété $\Rightarrow$ Automate Observateur



- Spécification dans le même langage

Observateurs

⋆ Propriété ⇒ Automate Observateur

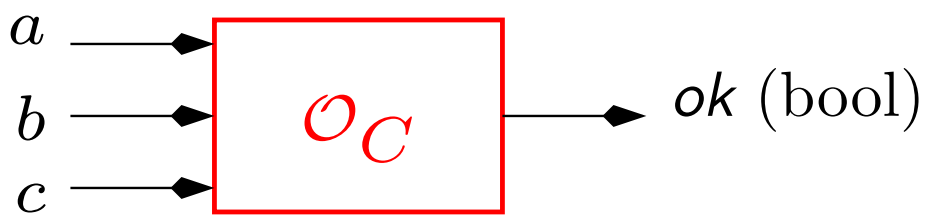


⋆ Spécification dans le même langage

⋆ Exécution en parallèle

Observateurs

⋆ Propriété ⇒ Automate Observateur



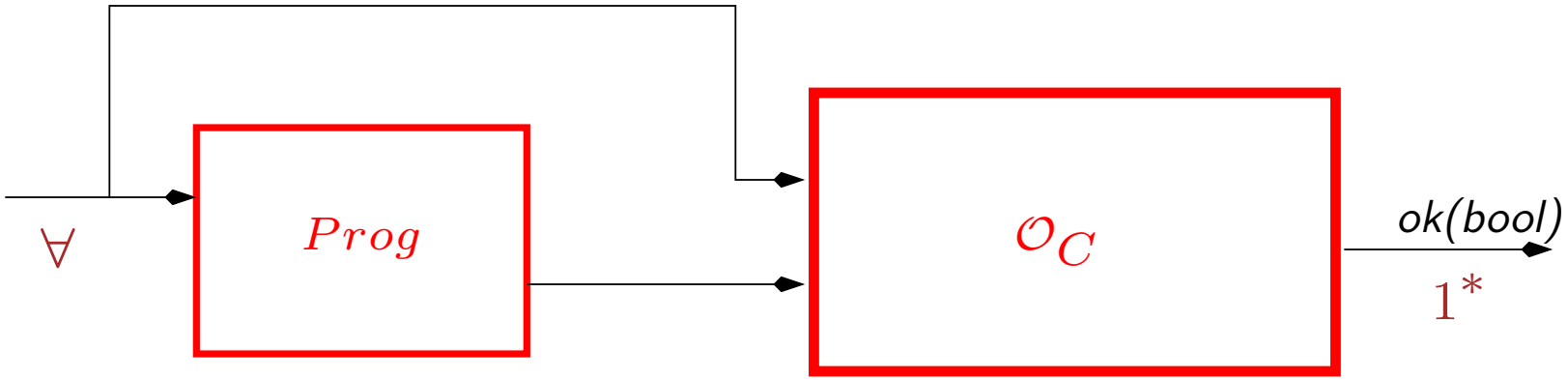
⋆ Spécification dans le même langage

⋆ Exécution en parallèle

➥ But : prouver $L(\neg C) \cap L(Prog) = \emptyset$

Observateurs - 2

Schéma :



Autour de Lustre

Des outils de vérification :

- Lésar : contrôle uniquement : méthode énumérative ou symbolique.

Autour de Lustre

Des outils de vérification :

- ⋆ Lésar : contrôle uniquement : méthode énumérative ou symbolique.
- ⋆ NBac : contrôle + approximation supérieure des états atteignables des variables numériques (polyèdres convexes).

Autour de Lustre

Des outils de vérification :

- ⋆ Lésar : contrôle uniquement : méthode énumérative ou symbolique.
- ⋆ NBac : contrôle + approximation supérieure des états atteignables des variables numériques (polyèdres convexes).

➥ Utilisation des outils existants

Une logique temporelle

"Quantified discrete duration calculus" [Pandya] :

⋆ Une logique temporelle

Une logique temporelle

"Quantified discrete duration calculus" [Pandya] :

- ⋆ Une logique temporelle
- ⋆ Une logique d'intervalles

Une logique temporelle

"Quantified discrete duration calculus" [Pandya] :

- ⋆ Une logique temporelle
- ⋆ Une logique d'intervalles
- ⋆ Modèles : suites finies de fonctions booléennes

Une logique temporelle

"Quantified discrete duration calculus" [Pandya] :

- ⋆ Une logique temporelle
- ⋆ Une logique d'intervalles
- ⋆ Modèles : suites finies de fonctions booléennes

➥ But : Construire $\mathcal{O}_\varphi$

QDDC - Connecteurs

Définition :

$$D ::= \lceil P \rceil^0 \mid \lceil\lceil P \rceil\rceil \mid \eta \; op \; c \mid \Sigma P \; op \; c \mid D_1 \wedge D_2 \mid D_1 \vee D_2$$

$$\mid D_1 \frown D_2 \mid \neg D$$

QDDC - Connecteurs

Définition :

$$D ::= \lceil P \rceil^0 \mid [\![P]\!] \mid \eta \; op \; c \mid \Sigma P \; op \; c \mid D_1 \wedge D_2 \mid D_1 \vee D_2$$

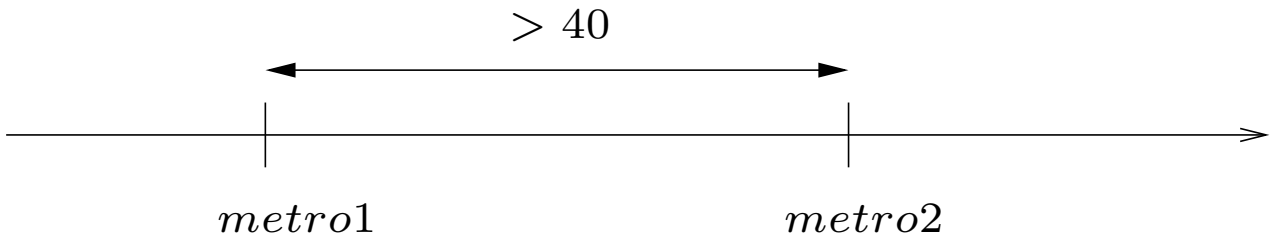
$$\mid D_1 \frown D_2 \mid \neg D$$

Raccourcis :

- $\Diamond D \equiv \texttt{true} \frown D \frown \texttt{true}$ (inévitablement D),
- $\Box D \equiv \neg \Diamond \neg D$.

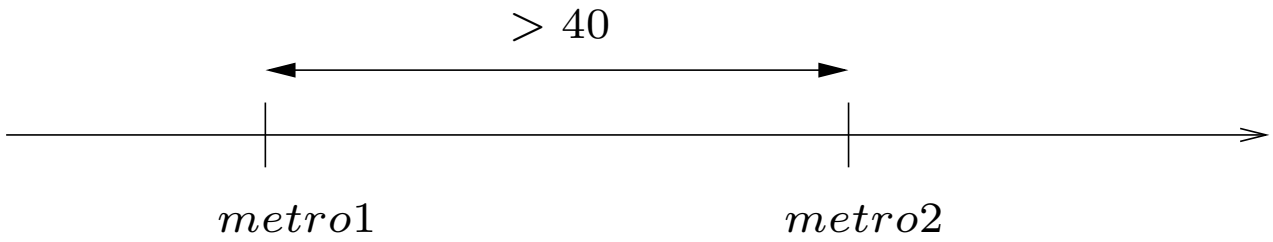
Exemples de propriétés

$\star\ \Box\Big(\big(\lceil metro1\rceil^0 \frown \eta \leq 40\big) \Rightarrow [\![\neg metro2]\!]\Big)$:

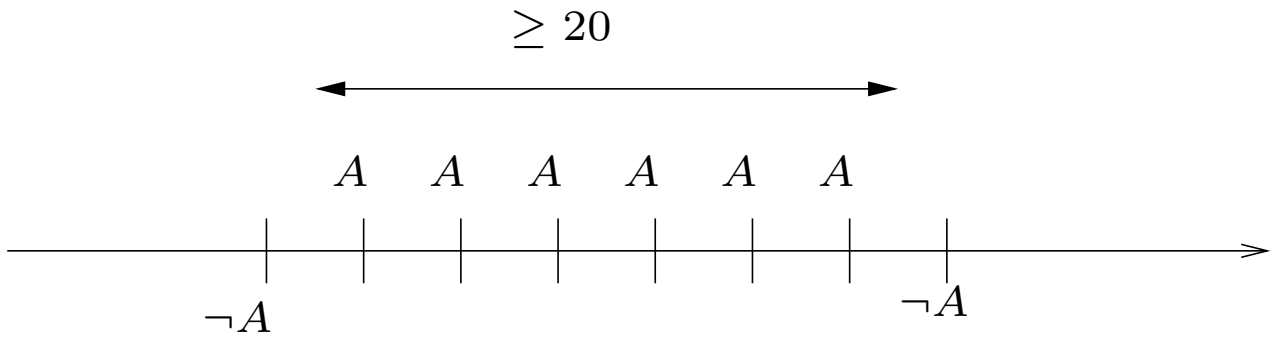


Exemples de propriétés

$\star\ \Box\Big(\big(\lceil metro1\rceil^0 \frown \eta \leq 40\big) \Rightarrow [\![\neg metro2]\!]\Big)$:



$\star\ \Box\Big(\big((\lceil \neg - Alarm\rceil^0 \wedge \lceil Alarm\rceil^0) \frown (\eta < 20)\big) \Rightarrow [\![Alarm]\!]\Big)$:



Exemples simples d'observateurs

⋆ Soit $\varphi_1 = \lceil\lceil p \rceil\rceil$:

```
ok = p -> p and pre(ok)
```

Exemples simples d'observateurs

- Soit $\varphi_1 = \lceil\lceil p \rceil\rceil$:

```
ok = p -> p and pre(ok)
```

- Soit $\varphi_2 = \Sigma P \leq \delta$:

```
nb_p = 0-> (if p then pre(nb_p)+1 else pre(nb_p));
ok = nb_p <= delta
```

Exemples simples d'observateurs

⋆ Soit $\varphi_1 = \lceil\lceil p \rceil\rceil$:

```
ok = p -> p and pre(ok)
```

⋆ Soit $\varphi_2 = \Sigma P \leq \delta$:

```
nb_p = 0-> (if p then pre(nb_p)+1 else pre(nb_p));
ok = nb_p <= delta
```

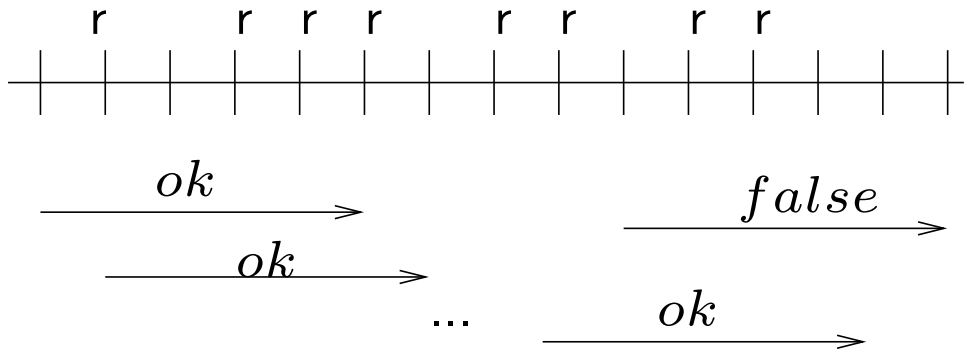
➥ Automatisons tout ça !

Problèmes

Soit $\varphi_3 = \Box\big((\eta > c) \Rightarrow (\Sigma r \geq d)\big)$:

Problèmes

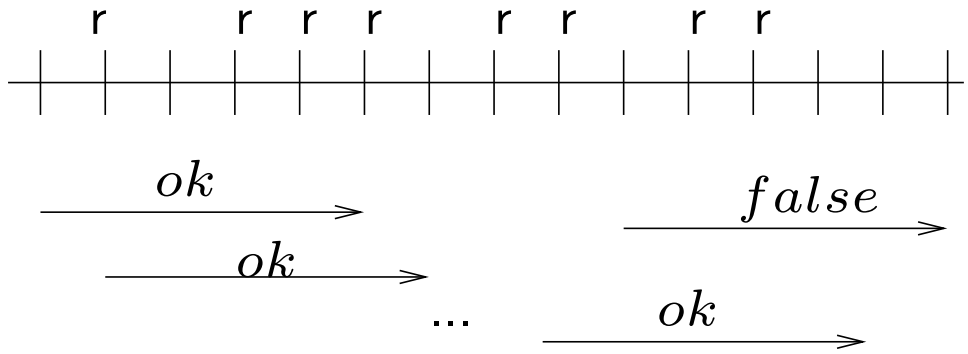
Soit $\varphi_3 = \Box\big((\eta > c) \Rightarrow (\Sigma r \geq d)\big)$:



⋆ détecter l'intervalle d'étude

Problèmes

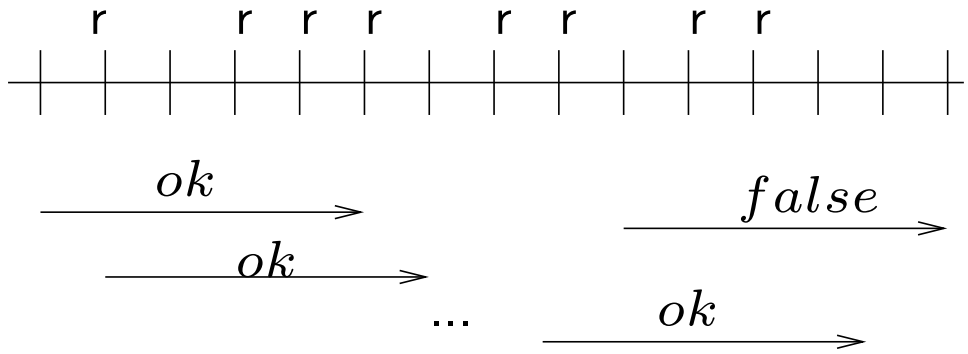
Soit $\varphi_3 = \Box\big((\eta > c) \Rightarrow (\Sigma r \geq d)\big)$:



- ⋆ détecter l'intervalle d'étude
- ⋆ déclencher au pire $2d$ compteurs

Problèmes

Soit $\varphi_3 = \Box\big((\eta > c) \Rightarrow (\Sigma r \geq d)\big)$:



⋆ détecter l'intervalle d'étude

⋆ déclencher au pire $2d$ compteurs

➥ Problème si d grand ou symbolique !

Solution déterministe

⋆ Solution radicale : Supprimer le “chop” !

Solution déterministe

- ⋆ Solution radicale : Supprimer le “chop” !
- ⋆ MAIS pouvoir d’expression diminué

Solution déterministe

⋆ Solution radicale : Supprimer le “chop” !

⋆ MAIS pouvoir d’expression diminué

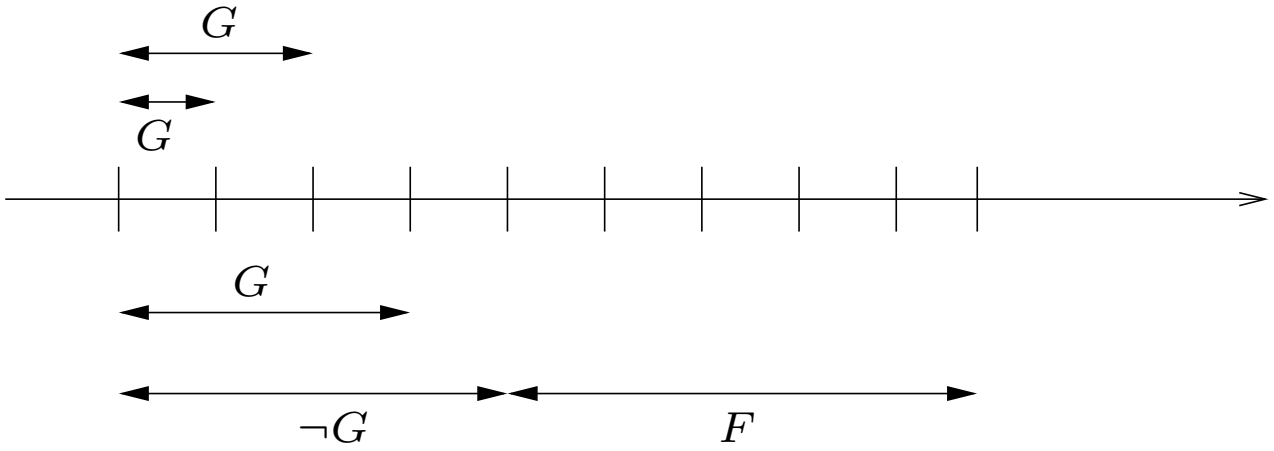
↪ Création du connecteur then.

Solution déterministe

⋆ Solution radicale : Supprimer le “chop” !

⋆ MAIS pouvoir d’expression diminué

➥ Création du connecteur then.



Solution déterministe - Fragment

Définition

$$G ::= \mathsf{begin}(P) \mid \lceil\lceil P \rceil\rceil \mid \eta \leq c \mid \Sigma P \leq c \mid \mathsf{age}(P) \leq c$$

$$\mid G \wedge G \mid G \vee G$$

Solution déterministe - Fragment

Définition

$$G ::= \mathsf{begin}(P) \mid \lceil\lceil P \rceil\rceil \mid \eta \leq c \mid \Sigma P \leq c \mid \mathsf{age}(P) \leq c$$

$$\mid G \wedge G \mid G \vee G$$

$$F ::= G \mid \mathsf{end}(P) \mid G \ \mathsf{then} \ F \mid F_1 \wedge F_2 \mid \neg F$$

Solution non-déterministe

- Déclenchement non-déterministe de la vérification

Solution non-déterministe

- ⋆ Déclenchement non-déterministe de la vérification
- ⋆ Ajout de nouvelles entrées : oracles

Solution non-déterministe

- ⋆ Déclenchement non-déterministe de la vérification
- ⋆ Ajout de nouvelles entrées : oracles
  - ➥ Automates non déterministes / ∀-automata

Solution non-déterministe - Exemple

$$\varphi_3 = \Box\big((\eta > c) \Rightarrow (\Sigma r \geq d)\big) :$$

Solution non-déterministe - Exemple

$$\varphi_3 = \Box\big((\eta > c) \Rightarrow (\Sigma r \geq d)\big) :$$

```
node phi3(r,k:bool) returns (ok:bool);
let
     assert (unique(k));
     length = if k then 1 else pre(length)+1;
     nb_r = if k then (if r then 1 else 0) else
                  if r then (pre(nb_r)+1) else pre(nb_r);
     ok = not after(k) or (true -> pre(ok)
                           and (length <=c or nb_r >=d));
tel
```

➥ On doit démontrer : quel que soit k

Utilisation des fragments

⋆ Outils : pour toute entrée

Utilisation des fragments

- ⋆ Outils : pour toute entrée
- ⋆ Variables d'observateur quantifiées universellement

Utilisation des fragments

- ⋆ Outils : pour toute entrée
- ⋆ Variables d'observateur quantifiées universellement

  ↪ Fragment indéterministe non fermé par négation

Utilisation des fragments

- ⋆ Outils : pour toute entrée
- ⋆ Variables d'observateur quantifiées universellement

  ↪ Fragment indéterministe non fermé par négation

Solution non-déterministe - Fragment

Définition

$$E ::= \lceil P \rceil^0 \mid \lceil\lceil P \rceil\rceil \mid \eta \; op \; c \mid \Sigma P \; op \; c \mid \neg E,$$

Solution non-déterministe - Fragment

Définition

$$E ::= \lceil P \rceil^0 \mid \lceil\lceil P \rceil\rceil \mid \eta \; op \; c \mid \Sigma P \; op \; c \mid \neg E,$$

$$D ::= E \mid D_1 \wedge D_2 \mid D_1 \vee D_2 \mid \exists p D \mid D_1 \frown D_2$$

Solution non-déterministe - Fragment

Définition

$$E ::= \lceil P \rceil^0 \mid \lceil\lceil P \rceil\rceil \mid \eta \; op \; c \mid \Sigma P \; op \; c \mid \neg E,$$

$$D ::= E \mid D_1 \wedge D_2 \mid D_1 \vee D_2 \mid \exists p D \mid D_1 \frown D_2$$

$$\text{QDDC-INDET} ::= \neg D$$

Conclusion

- Travail réalisé : identification des deux fragments QDDC-DET et QDDC-INDET

Conclusion

- ⋆ Travail réalisé : identification des deux fragments QDDC-Det et QDDC-Indet
- ⋆ Traduction prouvée et implémentée

Conclusion

- ⋆ Travail réalisé : identification des deux fragments QDDC-Det et QDDC-Indet
- ⋆ Traduction prouvée et implémentée
- ⋆ Études de cas :

Conclusion

- ⋆ Travail réalisé : identification des deux fragments QDDC-Det et QDDC-Indet
- ⋆ Traduction prouvée et implémentée
- ⋆ Études de cas :
  - expressivité bonne (Mine Pump)

Conclusion

- ⋆ Travail réalisé : identification des deux fragments QDDC-DET et QDDC-INDET
- ⋆ Traduction prouvée et implémentée
- ⋆ Études de cas :
  - expressivité bonne (Mine Pump)
  - vérification de propriétés paramétrées.

Conclusion

- ⋆ Travail réalisé : identification des deux fragments QDDC-DET et QDDC-INDET
- ⋆ Traduction prouvée et implémentée
- ⋆ Études de cas :
  - expressivité bonne (Mine Pump)
  - vérification de propriétés paramétrées.
  - vérification pas toujours concluante

Perspectives

- ⋆ Fragments maximaux

Perspectives

- ⋆ Fragments maximaux
- ⋆ Automates à oracles pour d'autres logiques

Perspectives

- ⋆ Fragments maximaux
- ⋆ Automates à oracles pour d'autres logiques
- ⋆ Meilleurs outils pour les variables numériques

Perspectives

- ⋆ Fragments maximaux
- ⋆ Automates à oracles pour d'autres logiques
- ⋆ Meilleurs outils pour les variables numériques

➥ Poursuite du travail en thèse

Et enfin

http://www-verimag.imag.fr/~danthony/