From Basic LOTOS To Description Logics

S. Coudert[(1)] L. Apvrille[(1)] C. Leduc[(2)]

(1) LabSoC, Telecom-ParisTech

(2) INRIA Grenoble

{sophie.coudert,ludovic.apvrille}@telecom-paristech.fr
chan.leduc@inrialpes.fr

Rencontre SAFA, 6 Mars 2008

DL(s): Description Logics

- Fragments décidables du premier ordre. Syntaxe dédiée.
  - Variantes: "kits" de constructions compatibles
  - Algorithmes théoriques, quelques-uns implantés
  - Pire des cas, NP-complet. Utilisables en pratique (IA)

- Variante utilisée: combinaison de fragments "connus".
  - Riche (expressive), décidable.
  - Algorithme à synthétiser (combinaison).
  - Complexité en pratique: reste à tester.

DL's Basics (variante utilisée)

Une "spécification" DL: Un langage et des axiomes

- **Langage**: $\mathbf{L} = (\mathbf{C}, \mathbf{N}, \mathbf{R})$
  - **C**: Concepts, i.e. relations unaires.
    N ($\subseteq$ **C**): concepts singletons.
  - **R**: Rôles, i.e relations binaires.
- **Axiomes**: $C \sqsubseteq C'$, $R \sqsubseteq R'$. Inclusion de concepts et de rôles.
  ($\equiv$: double inclusion)
- **Sémantique**: satisfiabilité usuelle. (décidable)
  Concept satisfiable: non vide dans un modèle.

Plus d'expressivité:

Constructions pour décrire des concepts et rôles complexes.

DL's Role and Concept Terms

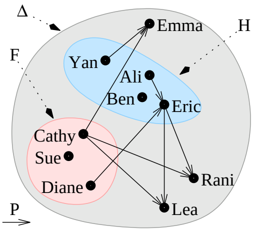
$\mathbf{L} = (\mathbf{C}, \mathbf{N}, \mathbf{R})$,
$\mathcal{I}$: interprétation. Δ: domaine de $\mathcal{I}$. $(C, D) \in \mathbf{C}^2$ $R \in \mathbf{R}$

- Concepts: (#: cardinal)
  $\top := \Delta$, $\bot : \emptyset$,
  $C \sqcap D : C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $C \sqcup D : C^{\mathcal{I}} \cup D^{\mathcal{I}}$, $\neg C : \Delta \backslash C^{\mathcal{I}}$,
  $\exists R.C : \{x \in \Delta \mid \exists y \in \Delta, ((x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$,
  $\forall R.C : \{x \in \Delta \mid \forall y \in \Delta, ((x, y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}})\}$,
  $\leq nR.C : \{x \in \Delta \mid \#\{y \in \Delta \mid ((x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\} \leq n\}$
  $\geq nR.C : \{x \in \Delta \mid \#\{y \in \Delta \mid ((x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\} \geq n\}$
- Rôles: (utilisables dans les concepts complexes)
  R^+: clôture transitive (sans réflexivité),
  R^-: relation inverse,
  $(R^-)^+$: combinaison des deux.

DL: Mini Exemple

$\mathbf{C} = \{$H: Homme adulte, F: Femme adulte$\}$. $\mathbf{R} = \{$ P: parent $\}$

Une interprétation

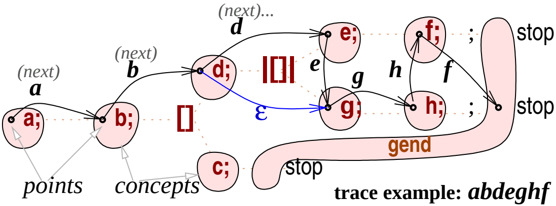


- *Rôles.*
  enfant: P^-,
  aïeul: P^+,
  descendant: $(P^-)^+$

- *Concepts.*
  parent: $\exists P.\top$,
  mère: $F \sqcap \exists P.\top$,
  orphelin: $\forall P^- \bot$,
  parent d'au moins 2 enfants: $\geq 2P.\top$,
  parent d'au plus 2 enfants: $\leq 2P.\top$,
  parent de 2 enfants: $= 2P.\top$,
  grand-père maternel: $H \sqcap \exists P.(F \sqcap \exists P.\top)$.

- *Axiomes.*
  Tout parent est adulte: $\exists P.\top \sqsubseteq F \sqcup H$
  On a au plus une mère: $\geq 2P.F \sqsubseteq \bot$
  On n'est pas son propre aïeul: complexe. $\Rightarrow$
  idée: père et mère fonctionnels surjectifs

Traduction: idées de base



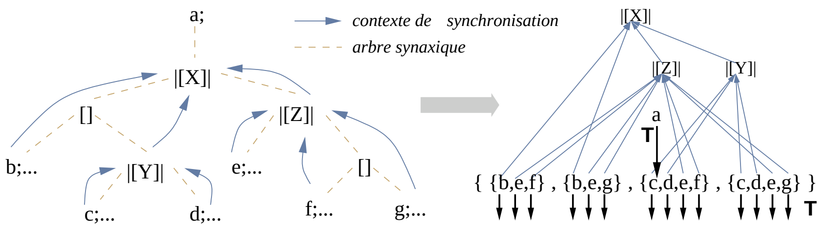
- **Axiomatisation** telle que "**interprétation** $\cong$ **une trace**"
- **Concepts: noeuds-portes** g de l'arbre syntaxique LOTOS.
  Points x de l'interprétation: points temporels de la trace.
  Si x est dans g, alors g est la prochaine porte franchie.
  $+\ g_{start}$ & g_{end}, concepts initiaux et finaux.
- **Rôles: actions LOTOS.**
  + "*next*", union des actions: successeur suivant la trace.
  + ϵ: mémorise les branches mises en attente (interleaving).
  + $\overrightarrow{g_{start}}$, action initiale.

Transformation du LOTOS Original

- Élimination des séquences
  ($>>$ et **exit** remplacés par synchronisation δ)
- Construction d'un "pseudo-graphe":
  - Les noeuds sont les portes
  - Ajout d'un point initial: g_{start}
  - "Pseudo-flêches": forêts.
    Succession par un unique opérateur:
    intégration des choix et parallélismes successifs
    $\rightarrow$ opérateur "choix entre ensembles de branches parallèles".
    Noeuds atteinds: premières portes des branches parallèles.
  - Déroulement fini des appels récursifs ($\Rightarrow$ rebouclage)
- Mémorisation des contextes de synchronisations

Portes, Succession et Synchronisation

Exemple: gstart; a; (b;... [] (c;...|[Y]| d;...)) |[X]| (e;... |[Z]| (f;...[] g;...))



- **A**: ensemble d'actions
- **G**: ensemble de portes
- **T**: $\mathbf{G} \rightarrow \mathcal{P}(\mathcal{P}(\mathbf{G}))$

  $g \mapsto$ alternative après g
- $sync(g)$: g est synchronisée

- SYNC(X). X: n portes d'une synchro. à n (statique. $n \geq 1$).
- SYNSET:$\{X \subseteq \mathbf{G} \mid SYNC(X)\}$

  Nécessaire: SYNSET fini

  $\Rightarrow$ Restrictions.

Problèmes, Déroulement Fini

main: a; P *et* P: P |[X]| b;...

$a \xrightarrow{T} \{\{b,b,b,...\}\}$ **!**

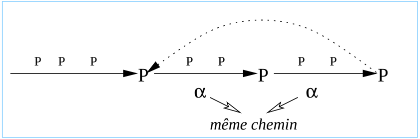
main: a; P |[X]| P *et* P: b;...

$a \xrightarrow{T} \{\{b,b\}\}$ **!**

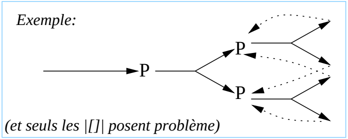
⇒ Récursion directe exclue.

⇒ Déroulement des appels récursifs *(Reboucle quand on retrouve les mêmes lieu d'appel, paramètres et contexte de synchro.)*

Solution actuelle: sûre et complexe



Mieux: dérouler 2 fois. Sûr(?)



⇒ Déroulement fini et nombre fini de points de synchronisation (SYNSET fini)

Axiomatisation: Idées

- Les axiomes contraignent les possibilités d'interprétation pour les rôles-action et les concepts-portes de sorte que "un modèle $\cong$ une trace"
- En résumé: On a un point initial (g_{start} singleton), et de chaque point, élément d'une porte g
  - une action $\vec{g}$ part vers une suite possible, i.e. un point d'une porte successeur selon $\mathbf{T}(g)$: la première d'une des branches parallèles (choix d'interleaving) d'un élement l'alternative des suites (choix LOTOS).
  - Des lien ϵ partent vers des points débutant les autres branches de l'alternative choisie: retours futurs par interleaving (toutes les branches, en cas de trace finie).
  - Les contraintes excluent toute transition à partir d'un état de blocage.
- $\Rightarrow$ Axiomes généraux + propriétés en chaque concept.

Axiomes Généraux

Langage: $\mathbf{L} = (\mathbf{G}^\bullet, \{g_{start}\}, \mathbf{A} \cup \{next, \epsilon\})$
Axiomes généraux:

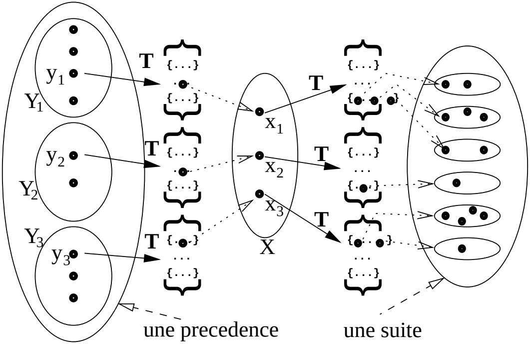
- La trace commence en g_{start} et n'y revient jamais
  $\{g_{start} \sqsubseteq \forall \overrightarrow{g}^{-}.\bot \mid \overrightarrow{g} \in \mathbf{A} \cup \{\epsilon\}\}$
  $\{g_{start} \sqsubseteq \bigsqcup_{g \in X / X \in \mathbf{T}(g_{start})} \exists \overrightarrow{g_{start}}.g\}$
- Chaque point (de la trace) est dans une porte et une seule.
  $\{\bigsqcup_{g \in \mathbf{G}^\bullet} g \equiv \top\}$
  $\{g \sqcap g' \sqsubseteq \bot \mid (g, g') \in (\mathbf{G}^\bullet)^2 \wedge g \neq g'\}$

Axiomes généraux, suite:

- *next* (représentant la trace) est une injection
  $\{\exists next.\top \sqsubseteq \leq 1 next.\top,\ \exists next^{-}.\top \sqsubseteq \leq 1 next^{-}.\top\}$
- La trace (*next*) contient toutes les actions et passe par tous les points pointés par un ϵ (interleaving).
  $\{\overrightarrow{g} \sqsubseteq next \mid \overrightarrow{g} \in \mathbf{A}\}$
  $\{\epsilon \sqsubseteq next^{+}\} \quad \cup \quad \{\exists\epsilon.\top \sqsubseteq \exists(next^{-})^{+}.g_{start}\}$
- Un point de porte non synchronisée est cible d'au plus un ϵ
  $\{g \sqcap \exists next^{-}.\top \sqsubseteq (\leq 1 \epsilon^{-}.\top) \mid g \in \mathbf{G} \wedge \neg sync(g)\}$
- La trace s'arrête après g_{end}.
  $\{g_{end} \sqsubseteq \forall next.\bot\}$

En un point: Précédences et Suites

Pour $X \in SYNSET$, $\mathbb{P}(X)$: précédences de X; $\mathbb{S}(X)$: suites de X



$$\mathbb{P}(X) = \{P \subseteq SYNSET \mid (\forall Y \in P, \exists x \in X, \exists y \in Y, \exists E \in \mathbf{T}(y), x \in y) \wedge (\forall x \in X, \exists! Y \in C, \exists y \in Y, \exists E \in \mathbf{T}(y), x \in y)\}$$

$$\mathbb{S}(X) = \{T \mid T = \bigcup_{g \in X} S_g \text{ where } \forall g \in X, (S_g \subseteq SYNSET \wedge \exists E \in \mathbf{T}(g), ((\forall Y \in S_g, \exists e \in E, e \in Y) \wedge (\forall e \in E, \exists! Y \in S_g, e \in Y)))\}$$

Axiomes: en un point accessible,...

Pour chaque $g \in \mathbf{G}$ (sans g_{end}), les "flêches" sortantes (actions, ϵ):

$$\{g \sqcap \exists next^{-}.\top \sqsubseteq \bigsqcup_{X \in SYNSET / g \in X \wedge \mathbb{S}(X) \neq \emptyset} \quad \text{(flêches vers } \mathbb{S}(X)\text{)}$$

$$\sqcup \bigsqcup_{X \in SYNSET / g \in X \wedge \mathbb{S}(X) = \emptyset} \quad \text{(flêches vers } g_{end}\text{)}$$

Deux cas de suite après le point de g considéré, selon qu'il est dans un point de synchronisation X avec ou sans suite possible.

La trace fait l'un des choix ouverts après X: elle sélectionne l'une des possibilités T de l'alternative décrite par $\mathbb{S}(X)$

Tous les ϵ vont vers T (ou g_{end}: traite le cas d'une trace finie)

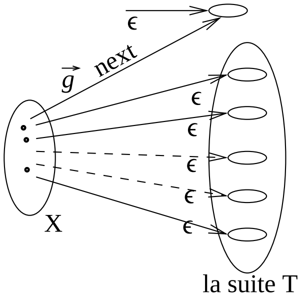
Et on a deux cas, selon que la trace modélisée continue sur la branche courrante ou "saute" ailleurs, réalisant un interleaving.

$$\bigsqcup_{T \in \mathbb{S}(X)} \Big(\forall \epsilon . (\bigsqcup_{flat(T) \cup \{g_{end}\}}) \quad \sqcap$$

$$\Big(\quad (\mathit{Cas\ d'une\ suite\ sur\ la\ même\ branche})$$

$$\sqcup \ (\mathit{Cas\ d'un\ saut\ par\ interleaving}) \quad \Big)$$

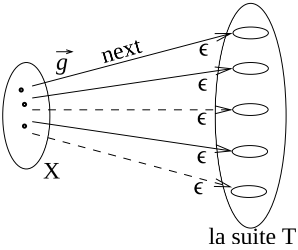
$$\Big)$$



On a une action vers la branche vers laquelle on saute et pour toutes les branches parallèles de T, un ϵ, obligatoire pour une trace finie (ligne 1) et potentiel pour une trace infinie (ligne 2).

$$\left(\exists \vec{g}.\exists\epsilon^{-}.\top \quad \sqcap \quad \bigsqcap_{Z \in T \cup \{\{g_{end}\}\}} (\exists\epsilon.\bigsqcup_Z \sqcap (\leq 1\epsilon \bigsqcup_Z)) \right)$$

$$\sqcup \quad \left(\exists \vec{g}.\exists\epsilon^{-}.\top \quad \sqcap \quad \bigsqcap_{Z \in T} (\leq 1\epsilon \bigsqcup_Z) \quad \sqcap \quad \forall next^{+}.\exists.next.\top \right)$$



On a une action vers le successeur sélectionné dans T par interleaving et pour toutes les autres branches parallèles, un ϵ (éventuel, si la trace est infinie).

$$\bigsqcup_{Y \in T} \left(\exists \vec{g}. \bigsqcup_Y \sqcap (\leq 1 \epsilon \bigsqcup_Y) \sqcap \bigsqcap_{Z \in T \cup \{\{g_{end}\}\}, Z \neq Y} (\exists \epsilon. \bigsqcup_Z \sqcap (\leq 1 \epsilon \bigsqcup_Z)) \right)$$

$$\sqcup$$

$$\bigsqcup_{Y \in T} \left(\exists \vec{g}. \bigsqcup_Y \sqcap (\leq 1 \epsilon \bigsqcup_Y) \sqcap \bigsqcap_{Z \in T, Z \neq Y} (\leq 1 \epsilon \bigsqcup_Z) \sqcap \forall next^+. \exists next. \top \right)$$

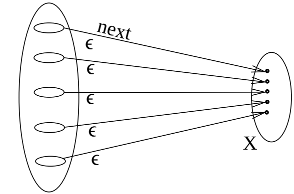
Axiomes: en un point de g, avec $g \in X \wedge \mathbb{S}(X) = \emptyset$

Si on est en un point de g dans le cadre d'un point de synchronisation X sans suite, on a trois cas:

- Saut (interleaving) & Terminaison différée (1 ϵ vers g_{end}).
- Trace infinie: on ne reviendra pas (pas de ϵ)
- Terminaison immédiate: action vers g_{end} et aucun ϵ.

$$\bigsqcup_{X \in SYNSET / \mathbb{S}(X)=\emptyset \wedge g \in X} \Big(\begin{array}{l} (\ \exists \overrightarrow{g}.\exists \epsilon^{-}.\top \ \sqcap\ \exists \epsilon.g_{end} \ \sqcap\ (\leq\ 1 \epsilon\, g_{end})) \\ \sqcup\ (\ \exists \overrightarrow{g}.\exists \epsilon^{-}.\top \ \sqcap\ \forall next^{+}.\exists.next.\top \ \sqcap\ \forall \epsilon.\bot\ ) \\ \sqcup\ (\ \exists \overrightarrow{g}.g_{end} \ \sqcap\ \forall \epsilon.\bot\ ) \quad \Big) \end{array}$$

Axiomes: en un point, le passé a eu lieu



Pour tout $g \in \mathbf{G}^{\bullet}$ (avec g_{end}), les "flêches" entrantes (actions, ϵ):

$$g \sqcap \exists next^{-}.\top \sqsubseteq \bigsqcup_{X \in SYNSET \mid g \in X} \left(\bigsqcup_{P \in \mathbb{P}(X)} \left(\bigsqcap_{Y \in P} (\exists \epsilon^{-}. \bigsqcup_{Y}) \right) \right)$$

Pour qu'un point soit accessible, il faut ququ'il soint dans un point de synchronisation X franchissable (sensibilisé), i.e. l'un des passés possible P avant X a eu lieu: un ϵ arrive de chacun des points de synchronisation de P.

Vérification: Idées

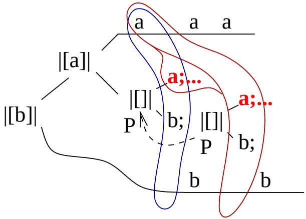
- Principe: (Axiomes LOTOS $\cup$ Axiomes Propriété) satisfiable?
- Propriétés à écrire à la main + Travail effectué sur des traductions automatiques de formules CTL (LTL) en DL
- Sémantique: Il existe un modèle $\cong$ Il existe une trace
- EFP, EGP se vérifie directement
- AGP, AFP se vérifie par l'absurde: vérifier $EF\neg P, EG\neg P$

$\Rightarrow$ Limite: imbrications de "always" / "exists" non vérifiables.

Amélioration possible

Points de synchronisation trop nombreux $\Rightarrow$ points accessibles

- Frontière après un point: on avance, faisant des choix lotos et prenant toutes les branches parallèles jusqu'à rencontrer des points synchronisés $\Rightarrow$ ensemble de points synchronisés. Après un ensemble de point: "union" pour tous les points.
- Algo: dans la frontière après $\{g_{start}\}$, **(*)** identifier les points de synchro contenus puis déclencher chacun d'entre eux: le remplacer ("union") dans la frontière par la frontière qui lui succède. itérer en **(*)**. (En fait , +complexe: après un point, une alternative de frontières).
- Limite: Condition d'arrêt = frontière déjà rencontrée ("union" $\Rightarrow$ nombre fini de frontières possibles). Fusion de points $\Rightarrow$ perte possible de points accessibles. (caractériser les cas qui marchent ou résoudre le problème...)



Problème: deux a fusionnés $\Rightarrow$ deux frontières confondues. Parfois sans conséquences, parfois si: la rouge est perdue et si la bleue ne la représente pas,. . .

- Conviction: c'est soluble. Sans génération d'un nombre infini de points de synchro, il existe un déroulement suffisant. "hide" plus problématique.
- Conviction étendue: construction d'un "automate" allant jusqu'aux points de rebouclages optimaux (accessibilité temporelle). Modèle sémantique intéressant.
- Limites DL: petit infini, une complexité qui perdure (low-level) $\Rightarrow$ Approches Logiques par preuves guidées à explorer.