Techniques pour le *model-checking* de spécifications de Haut-niveau

Yann Thierry-Mieg

Séminaire Méthodes de Conception, Vérification et Réalisation
Application à la Répartition et au Temps Réel
21 Janvier 2005

Introduction
Systèmes Répartis Fiables

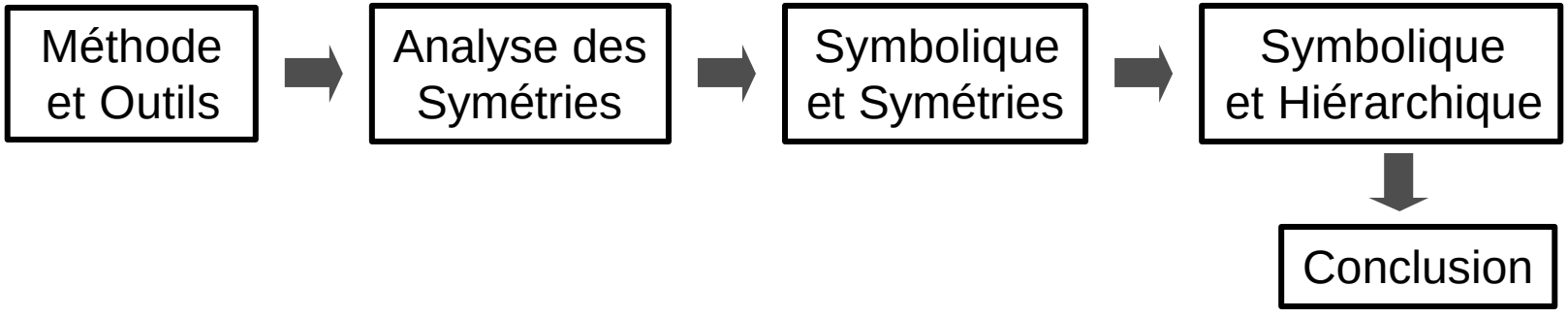
- **Crise Chronique du Logiciel**
  - **Taille, complexité des applications augmentent**
  - **Dépendance accrue, embarqué => besoin de qualité accrue**
  - **Cycle de vie réduit**
- **Besoin d'outils**
  - **Fonctionnement type compilateur [Hoare'03]**
  - **Fournissant la garantie que l'*application* "se comportera bien"**
- **Systèmes Répartis Complexes**
  - **Indéterminisme, Entrelacements, Interblocages, Famines**
- **Existant :**
  - **Test**
  - **Vérification formelle**
    - *Systèmes de preuve (preuve assistée)*
    - *Analyse Structurelle (automatique, limitée)*
    - *Model-Checking (automatique)*

Model Checking de Systèmes Répartis

- **Problèmes :**
  - **Explosion combinatoire de l'espace d'états**
  - **Intégration des méthodes au processus de développement**
- **Espaces d'états:**
  - **Graphes réduits**
    - *Symétries* → Réseaux de Petri Bien Formés [CDFH'90]
    - *Ordre Partiel*
  - **Approches par Compression**
    - *Symbolique (BDD)* → Data Decision Diagram [CEPAPW '02]
    - *Compositionnelle* → Smart [MC'99,CMS'03]
    - *Etats, clés de hash, valeurs inutiles...*
- **Descriptions formelles et Développement Logiciel**
  - **Augmenter l'expressivité**
  - **Transformations de modèles** → Language for Prototyping [R'03,MORSE'03-06]

Plan

- **Méthodologie de Développement et Modèles formels**
- **Des techniques complémentaires pour lutter contre l'explosion combinatoire :**
  - **Analyse automatique des Symétries :**
    - *Exploitation par les Réseaux de Petri Bien Formés*
  - **Approche Symbolique et Symétries**
    - *"Symbolique-Symbolique"*
  - **Approche Symbolique et Compositionelle**
    - *Diagrammes de décision hiérarchique : Set Decision Diagram*



Méthodologie de Développement et Réseaux Bien Formés

Méthode et Outils → Analyse des Symétries → Symbolique et Symétries → Symbolique et Hiérarchique

Une Méthodologie par Transformations de Modèles

Intégrer le model-checking aux standards industriels

- Complémentaire d'une reflexion sur UML en cours

Model Driven Architecture [OMG]

- Modèle central
- Transformations des modèles

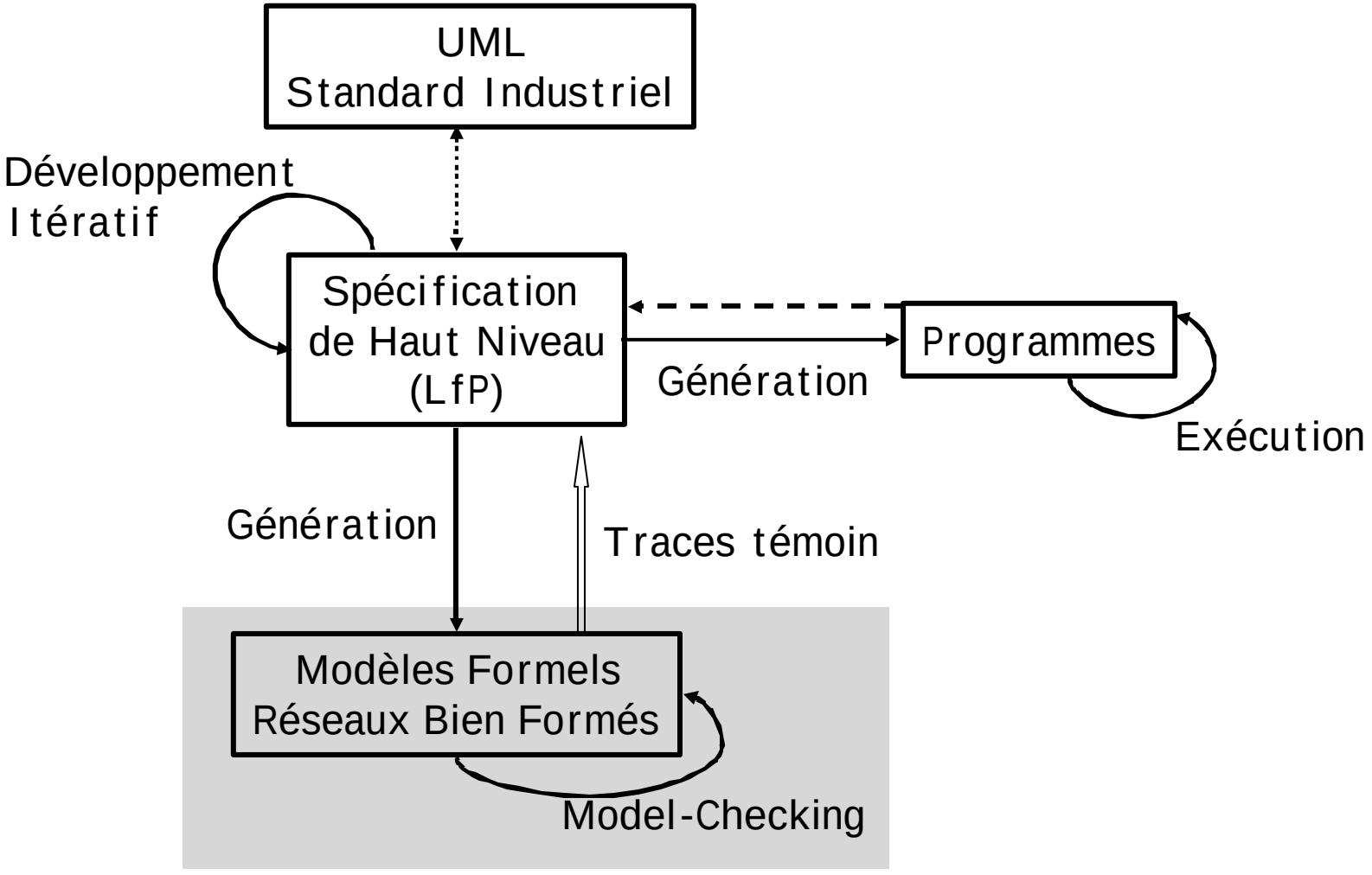
Notre Cible :

- Systèmes Répartis
  - *Model-checking et génération de code*

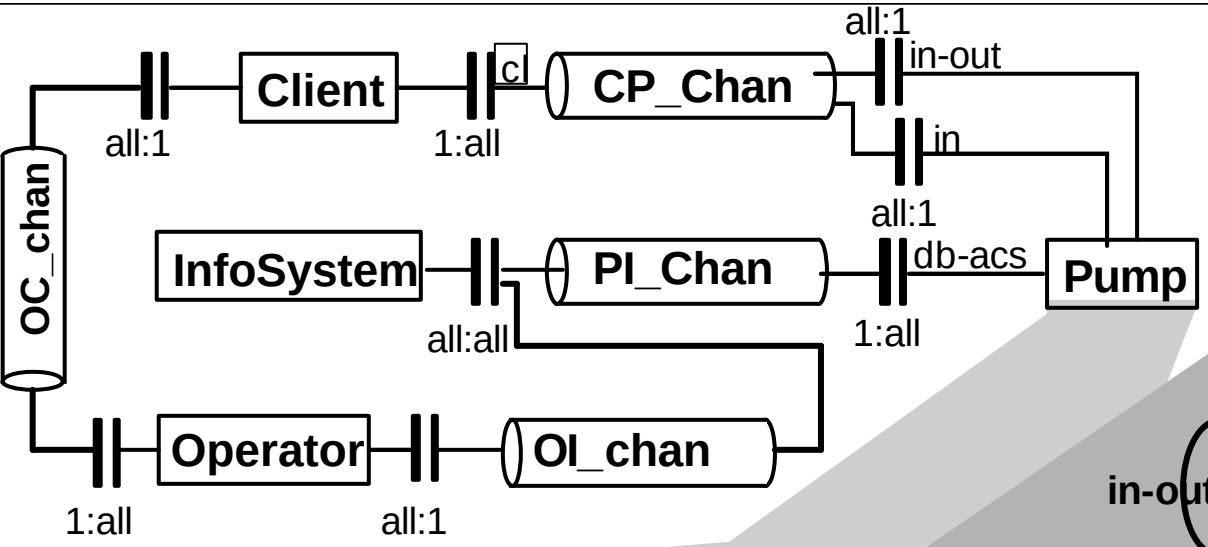
Contraintes sur le modèle central

- types finis, pas de récursion
- sémantique formellement définie
- mécanisme assurant la cohérence modèle/implémentation

Méthodologie de LfP
Language for Prototyping

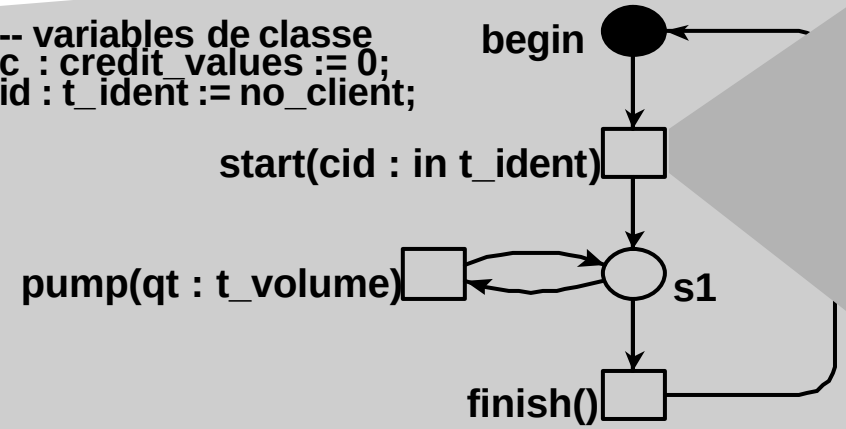


Un langage de Haut niveau (LfP)

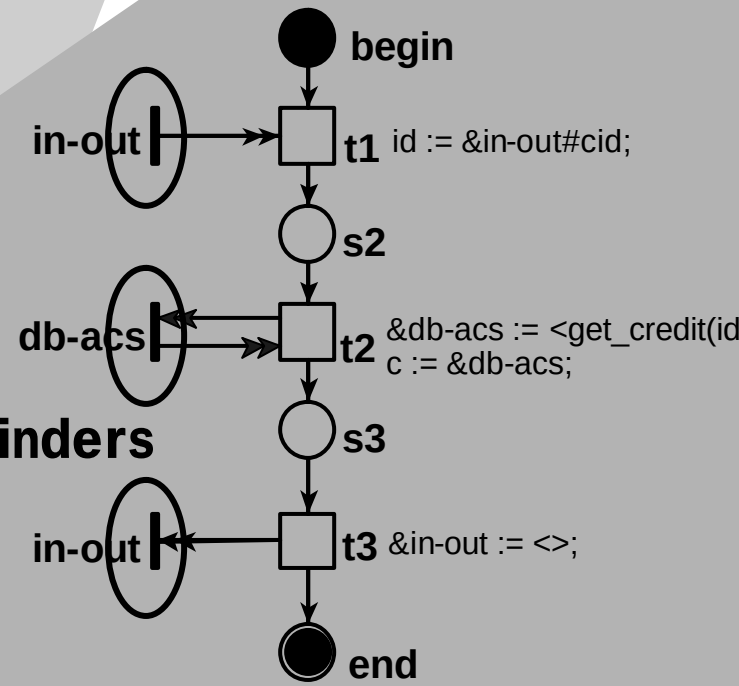


Définition de l'Architectu

Composants
Connecteurs
Topologie



Comportement d'un composant



Comportement d'une méthode

Transformation et Analyse

- **Tests "à la main"**
  - **Gros modèles**
  - **Passage à l'échelle ?**
- **Informations de haut niveau exploitables**
  - **Grands domaines**
  - **Instances répétées**
  - **Plusieurs grains de composition**
  - **Abstractions selon la propriété**
- **Développement de méthodes**
  - **Exploitant ces informations**
  - **Supportant le passage à l'échelle**

Choix des Réseaux Bien-Formés

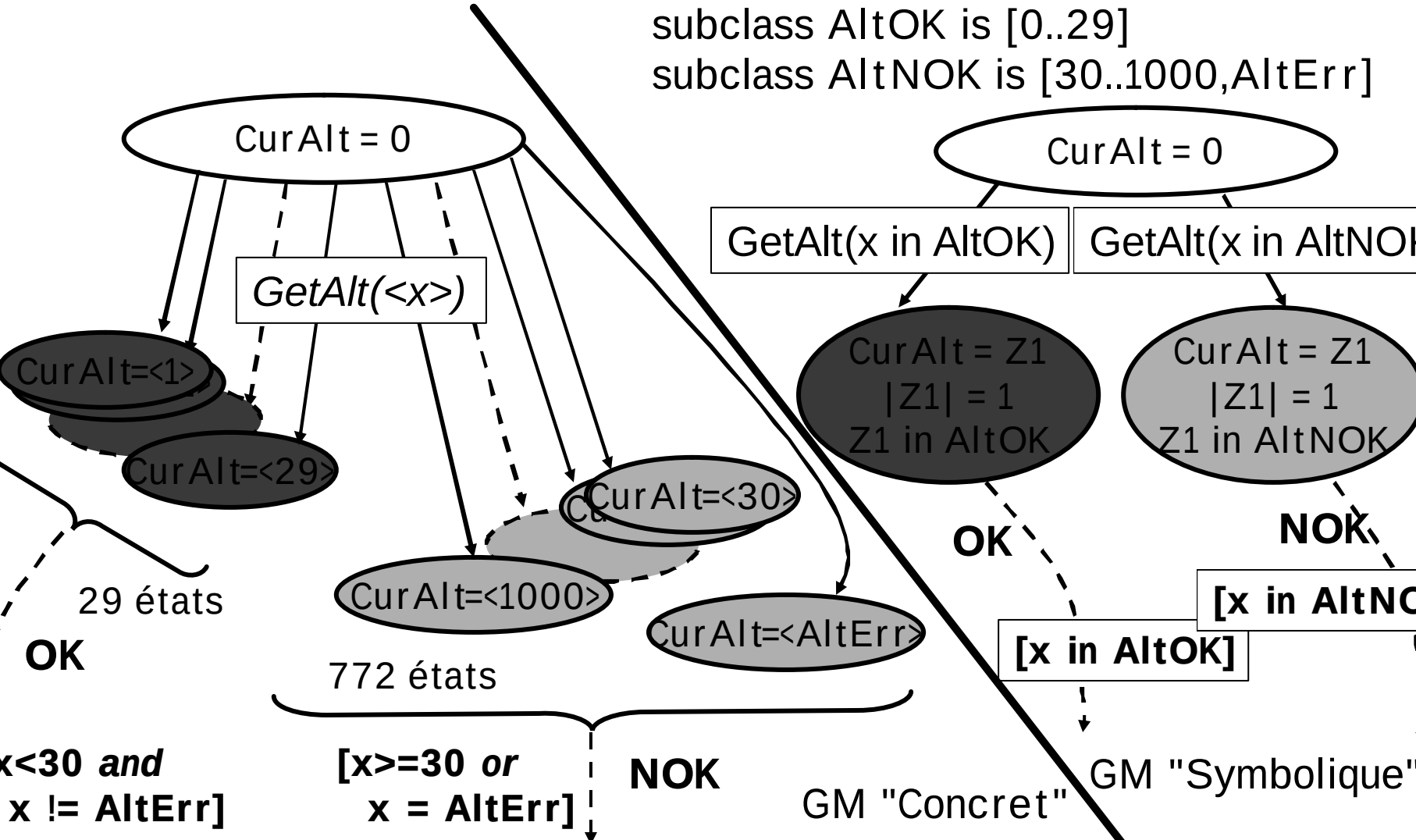
Symétries, Performances, Composition,

Paramètrés, Expressifs, Normalisés...

Constatation : Existence de symétries

- **Systèmes répartis:**
  - **Instances répétées : clients, processus ...**
  - **Domaines symétriques : adresses mémoires, pid, ...**
  - **Discrétisation de variables au domaine continu : capteurs...**
    - *Grands domaines mais valeurs critiques au contrôle : i < seuil, i ¹ 0 ...*
- **Exploitation des symétries**
  - **Réseaux de Petri Bien Formés [CDFH'90]**
  - **Murphi [IpDill'96]**
  - **Extensions symétries partielles [Capra'00,BHI'04]**
  - **...**
- **Graphe des Marquages Symboliques (GMS)**
  - **Basé sur des permutations qui présèrvent le comportement**
  - **Permutations décrites comme une partition**
  - **Un état symbolique = classe d'équivalence représentative du comportement**

Graphe de Marquages Symboliques (GMS)



Analyse automatique des symétries et Réseaux Bien Formés

Méthode et Outils → **Analyse des Symétries** → Symbolique et Symétries → Symbolique et Hiérarchique

Propriétés du Graphe des Marquages Symboliques (GMS)

Classes d'équivalence

- *pour la relation de franchissement*
- *permutations d'instances*

Gain potentiellement *exponentiel*

- **Nombre d'états symboliques très faible**
- **Franchissement symbolique**
- **Gain lié à la cardinalité des sous classes**

Problème

- **Formalisme contraignant**

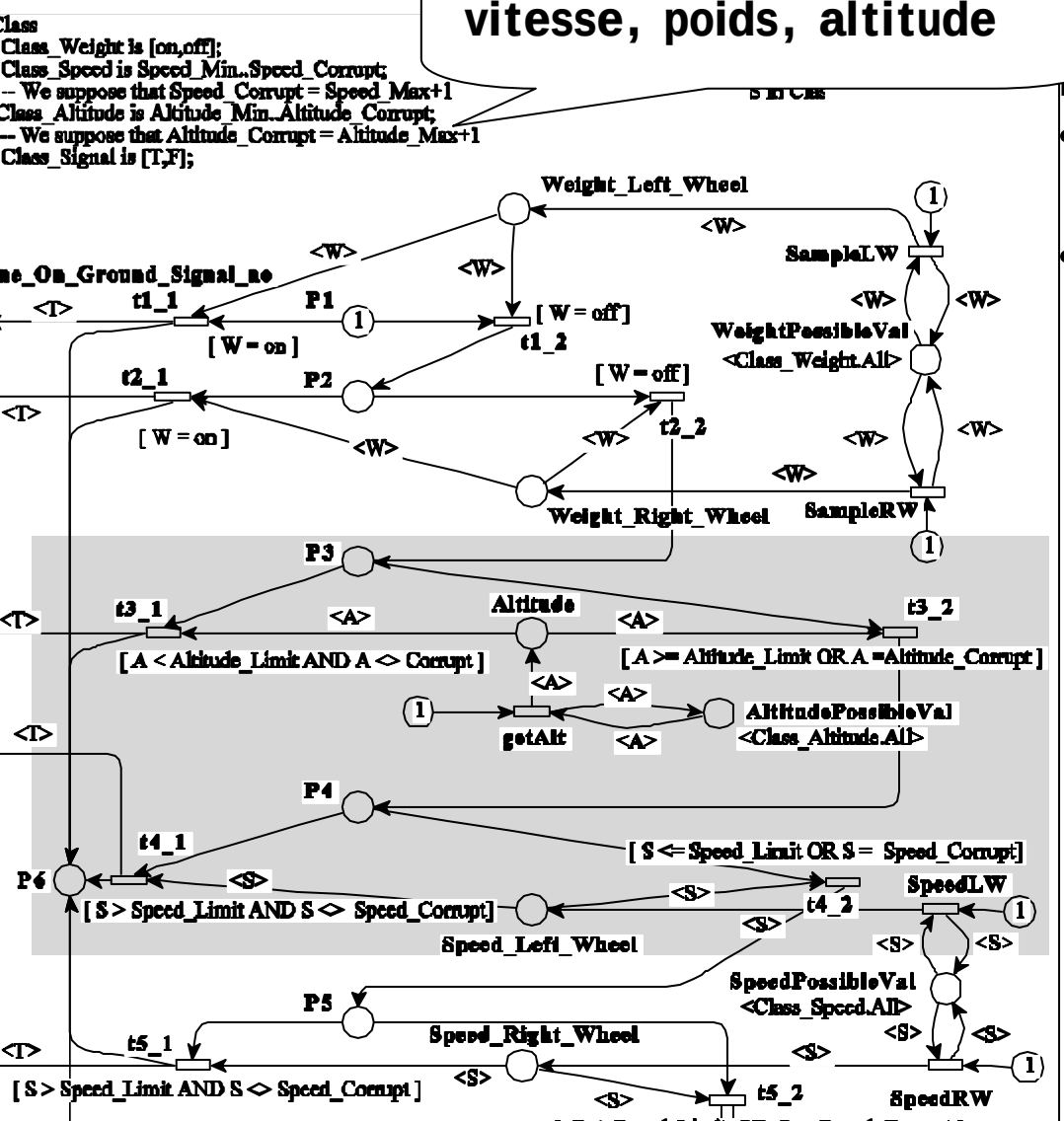
  ++expressivité --expression

- **Information absente sur langage de Haut-niveau**

Objectif : Transparence des Mécanismes

Analyse Automatique des Symétries

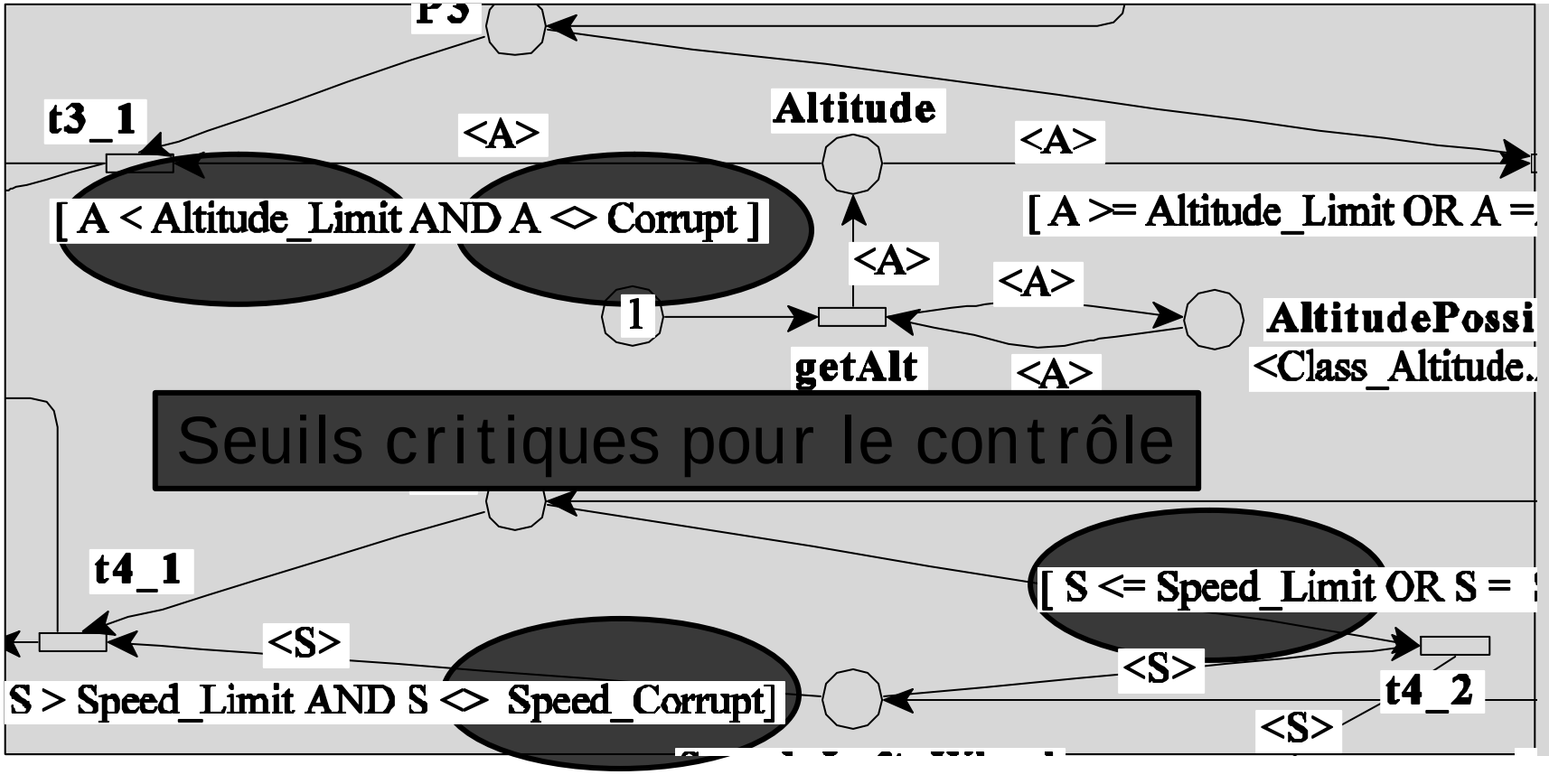
Grands domaines
vitesse, poids, altitude



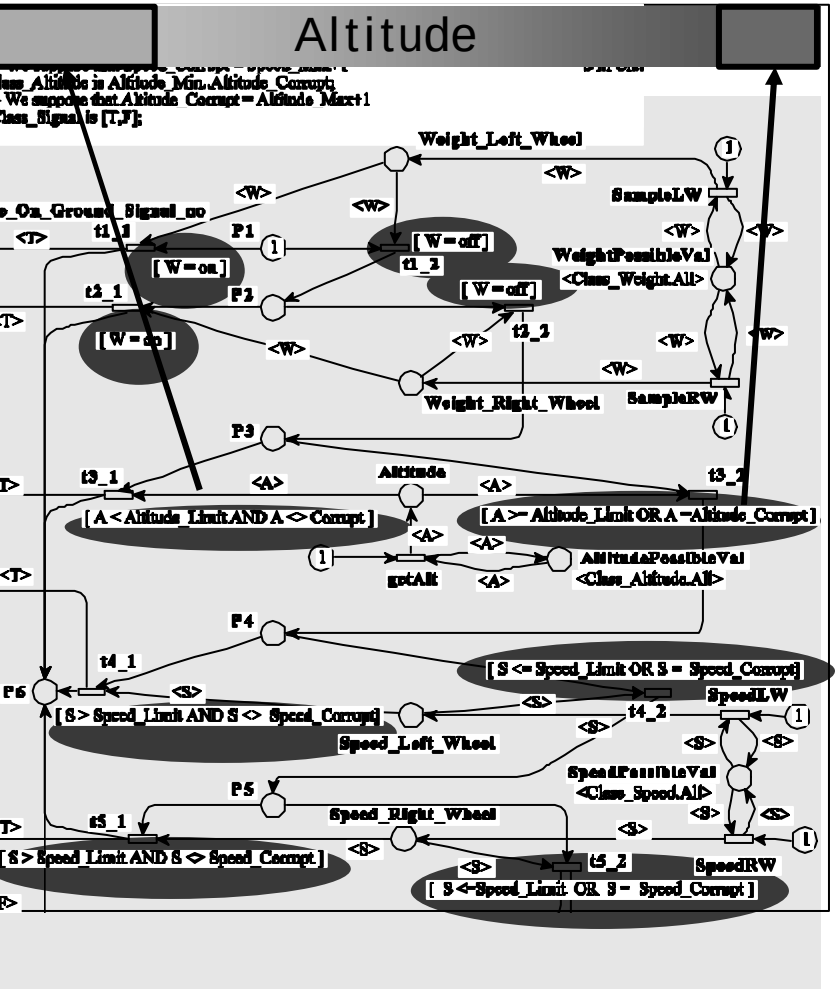
pécifications industrielles

- Instruments de Contrôle Avionique (Projet FORMA)
- Domaines discrétisés

Existence de seuils



Analyse Automatique des Symétries : Principes

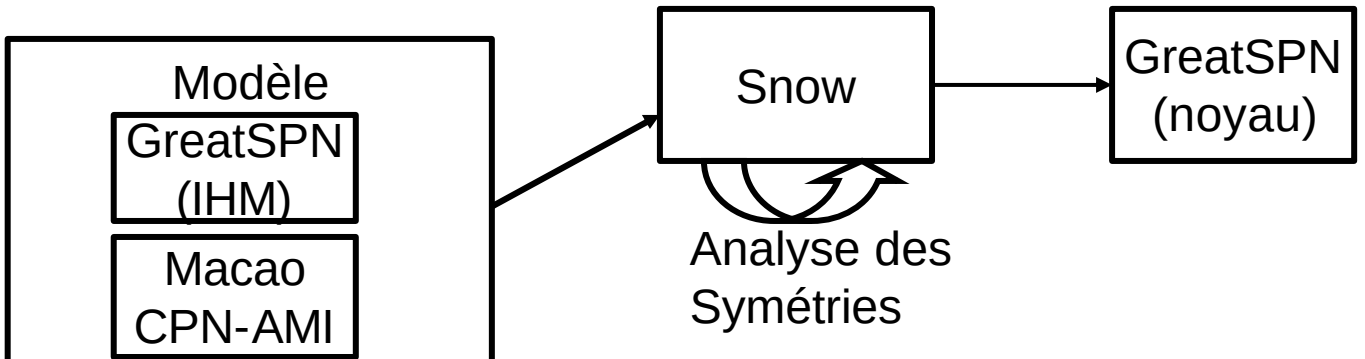


- **Extensions de syntaxe**
  - **Plus de mention explicite des sous classes**
- **Analyse :**
  - **Algorithmes séparés**
    - *Gardes, fonctions de couleur, marquage initial*
  - **Normalisation**
  - **Représentations canoniques**
    - *Exhibent les symétries [TMDM'03]*
- **Composition par raffinement**

Approche complètement automatisée

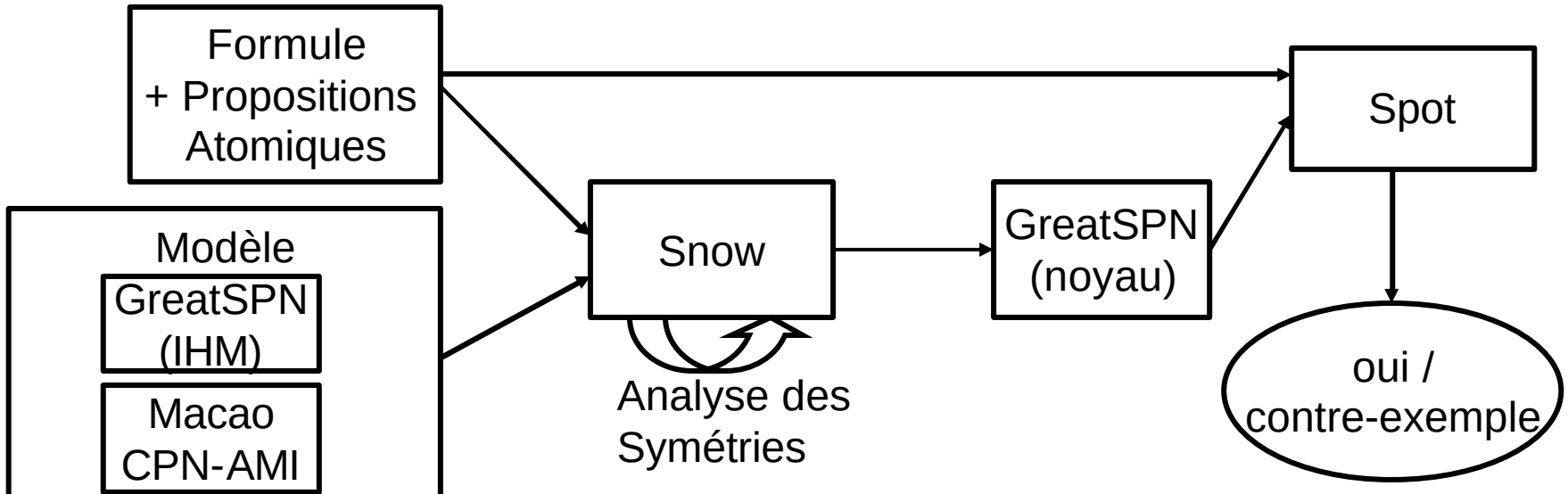
Analyse des symétries d'un modèle

- **Méthode Structurelle**
  - **Complexité linéaire sur la taille du modèle**
- **Définition *automatique* des symétries admissibles**
  - **Modélisation en langages de plus haut niveau**
  - **Utilisation *transparente* des méthodes symboliques**
  - **Pas de risque de sur-spécification**
  - ***lisibilité*, *modifications facilitées*,**
- **Contribue à la *diffusion* de ces techniques**



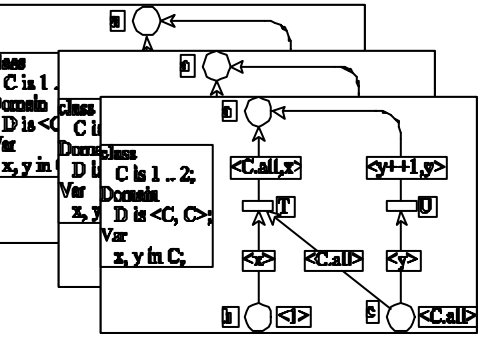
Analyse des symétries d'une propriété

- **Problème:**
  - **Vérification de propriété asymmétrique sur un GMS**
  - **relation d'équivalence GMS => pour la relation de franchissement**
  - **Propriété asymétrique => distinction d'instances**
- **Solution:**
  - **Analyse des symétries d'une formule**
  - **Adapter la relation d'équivalence du GMS**

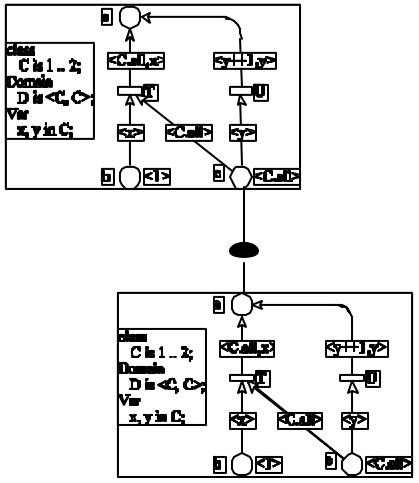


PolyOrb : Etude de Cas avec Snow

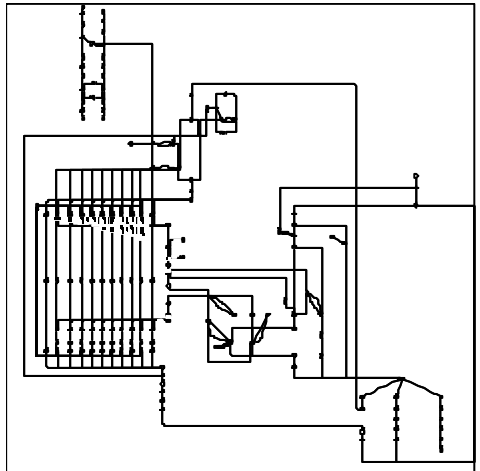
- **Modèle d'intergiciel "schizophrène"**
  - **séparation du contrôle et des données**
  - **vérification de propriétés fonctionnelles du coeur de PolyOrb = µBroker**
- **Méthode adoptée (semblable à LfP) :**
  - **Un composant -> plusieurs modèles (politiques possibles)**
  - **Composition des modèles -> une configuration du µBroker**



1. Librairie de modèles



2. Assemblage d'une configuration

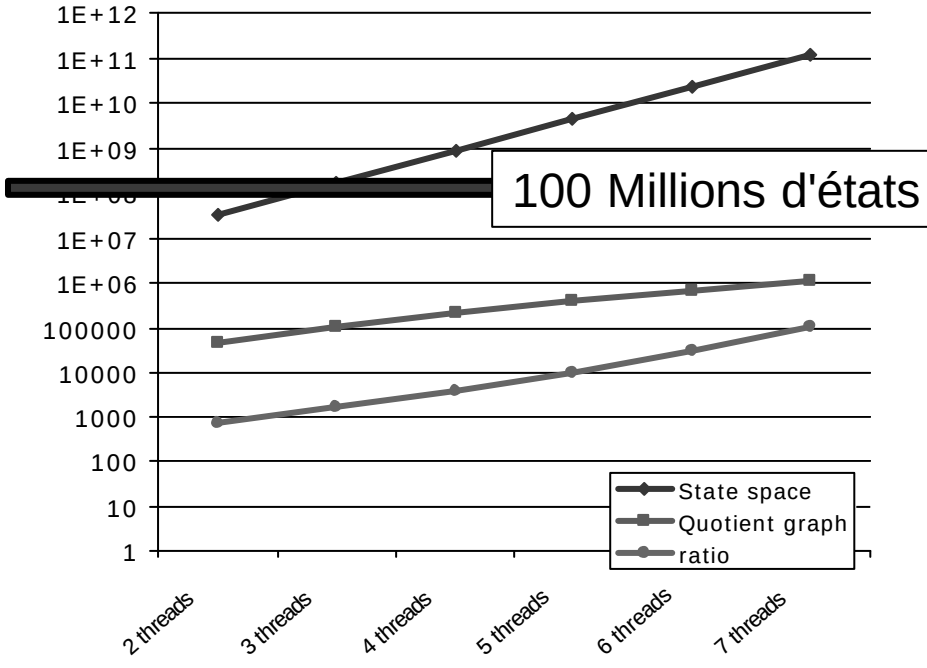


3. Analyse

Performances PolyOrb

(P4 2.4GHz 512Mo)

- **Le modèle est relativement gros**
  - **80 places, 240 arcs**
- **Modèle fortement symétrique**
  - **Désiré à la conception**
- **Analyse automatique permet :**
  - **vérification configurations diverses**
  - **adaptation aux propriétés**
- **Vérification de 4 propriétés pour une politique donnée**
  - **symétrie, cohérence, absence d'interblocage, équité**



- **La vérification devient possible**
  - **Echoue avec des méthodes classiques pour seulement 3 threads**

Approches Symboliques Symboliques

Méthode et Outils → Analyse des Symétries → **Symbolique et Symétries** → Symbolique et Hiérarchique

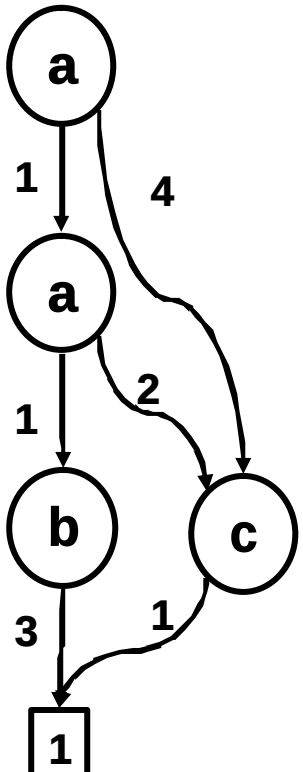
DDD: Data Decision Diagrams

- **Diagrammes de Décision:[BCM'92]**
  - **Initialement BDD [Bryant86]**
  - ***Structure compacte* pour représenter des *ensembles***
  - **Table d'unicité et cache d'opérations**
  - **Complexité liée au *nombre de noeuds***
  - **Exploite les *symétries implicites* entre les éléments de l'ensemble**
  - **Problème des pics intermédiaires**
- **Très grand succès**
  - **SMV, Smart, Uppaal, Prism, ...**
- **Data Decision Diagram [Couvreur+02]**
  - **variables entières, pas d'ordre, chemins longueur variable**
  - **Opérations ensemblistes + *homomorphismes inductifs***

$$a \xrightarrow{4} c \xrightarrow{1} 1$$

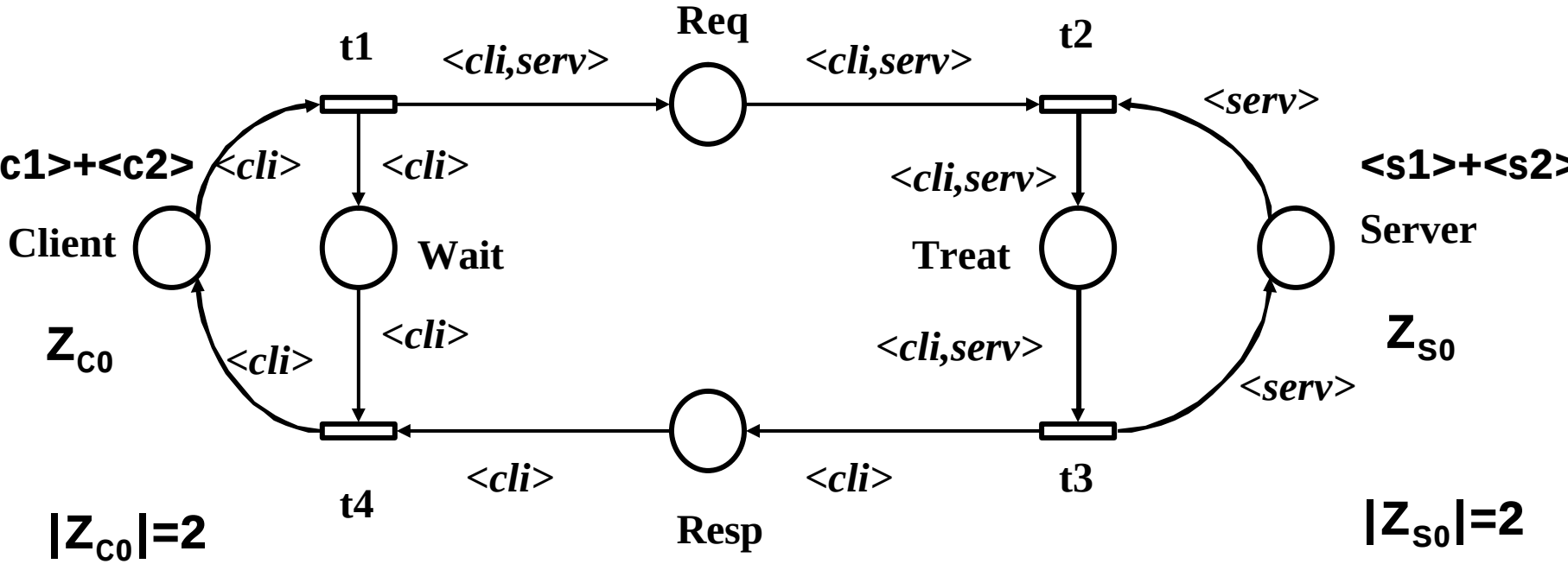
$$a \xrightarrow{1} a \xrightarrow{2} c \xrightarrow{1}$$

$$a \xrightarrow{1} a \xrightarrow{1} b \xrightarrow{3}$$



DDD et Réseaux Bien-Formés

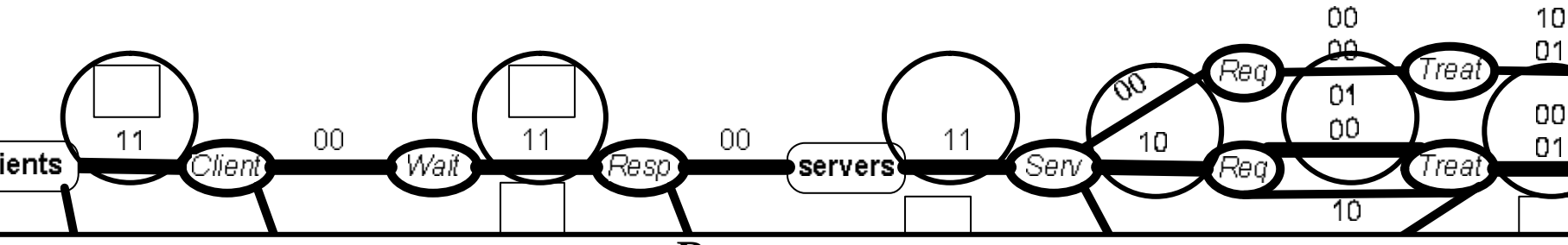
- **Objectif:**
  - **Représenter un ensemble d'états du *GMS dans un DDD***
  - **Définir le *franchissement symbolique* avec des *homomorphismes inductifs***

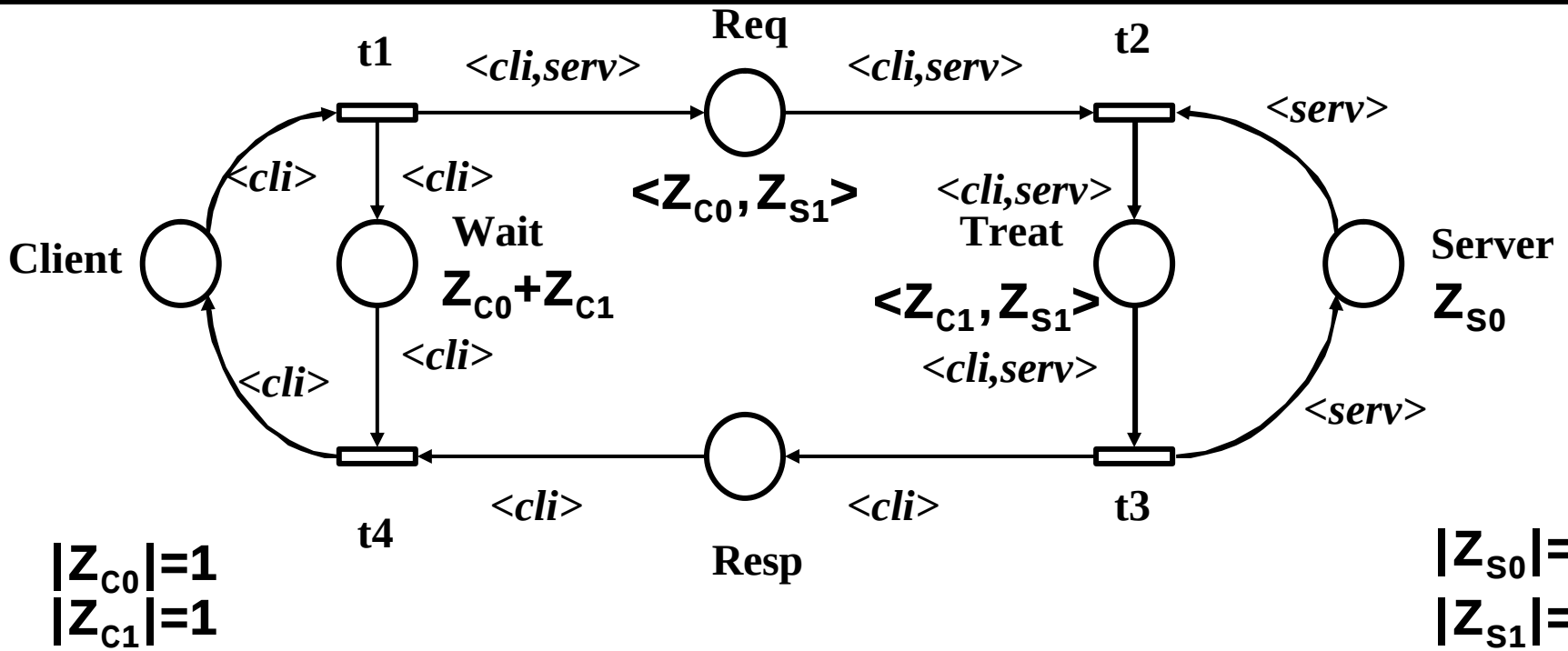


Marquage Concret

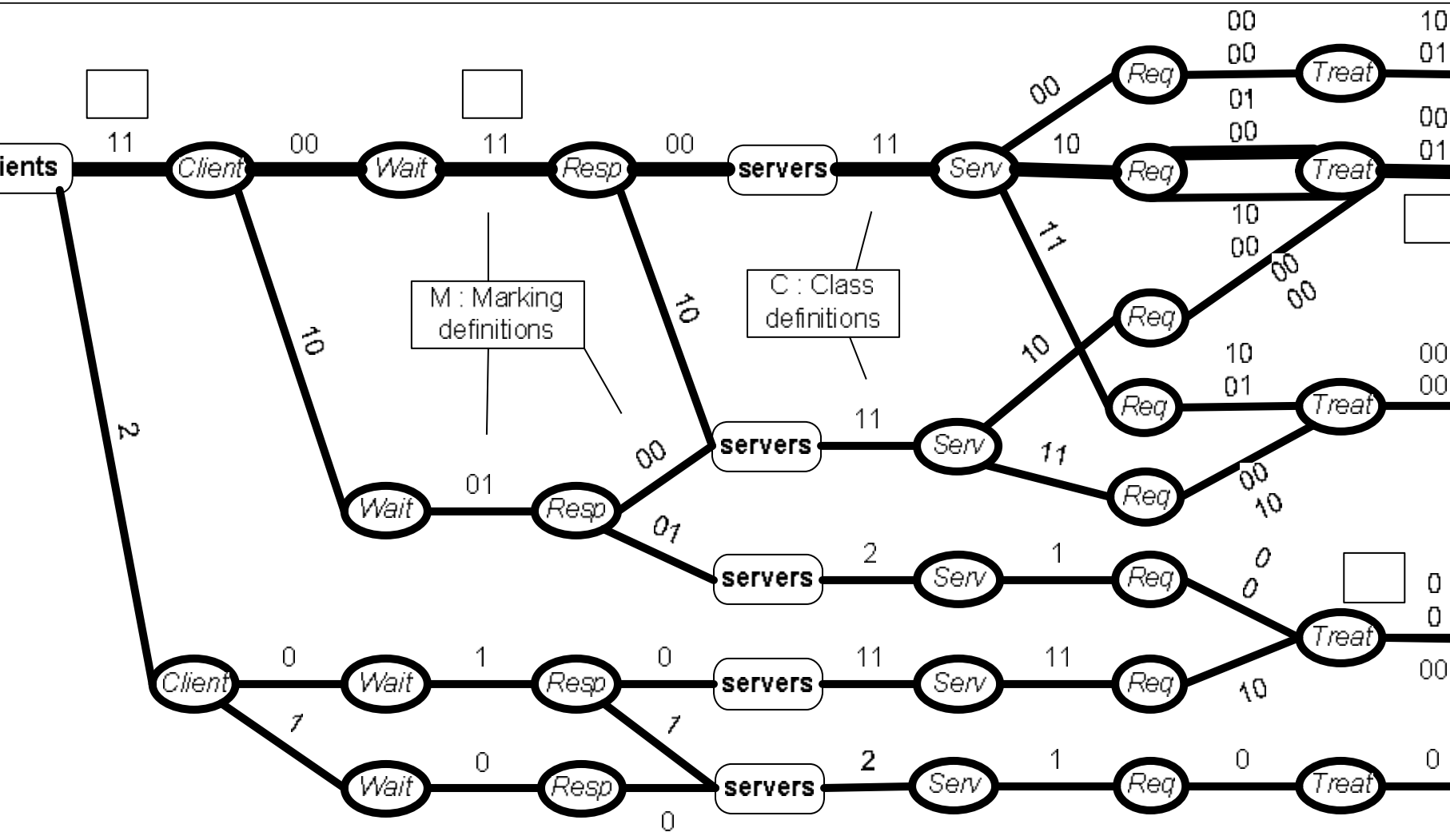
Marquage Symbolique

Représentation *symbolique* d'un *marquage symbolique*





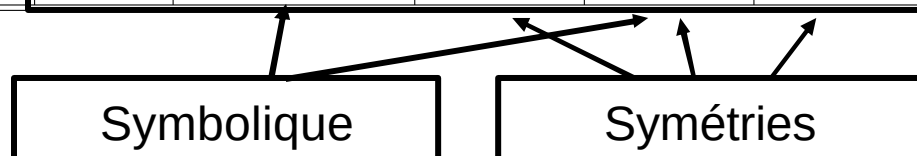
2 clients, 2 serveurs


ients
Client
Wait
Resp
servers
Serv
Req
Treat
M : Marking definitions
C : Class definitions
11
00
01
10
2
1
0

Performances du Prototype
(P4 2GHz 512Mo)

- **Excellente complexité mémoire**
  - **1500 noeuds, pour 10^620 états / 250k états symb.**
  - **Combine les gains mémoire des deux approches**
- **Complexité temporelle élevée**
  - **mais moins qu'avec des approches explicites (GreatSPN 3h20 cli/serv 9x9)**
  - **Problème de taille intermédiaire**
- **Résultats probants**

Modèle	N (#)		Etats (#)	GMS (#)	noeuds (#)	temps (sec)
Client Serveur	5	2	5484	82	884	3
	10	2	$1.35 \cdot 10^{7}$	476	2419	28.56
	20	2	$4.17 \cdot 10^{13}$	3201	2914	116.92
	6	6	$2.44 \cdot 10^{7}$	281	4091	21.2
	8	8	$1.12 \cdot 10^{11}$	964	13123	170.4
	9	9	$1.05 \cdot 10^{13}$	1698	22016	450.5
Peterson	7		692777	320	6159	15.7
	10		$3.46 \cdot 10^{9}$	3328	42442	247.0
	11		$7.19 \cdot 10^{10}$	7168	74039	545.4
	12		$1.62 \cdot 10^{12}$	15360	126807	1946.4
SC Vagues	40		$1.74 \cdot 10^{20}$	5620	831	14.83
	100		$1.76 \cdot 10^{49}$	35050	1011	127.88
	200		$1.79 \cdot 10^{97}$	140100	1313	1041.75
	300		$1.02 \cdot 10^{371}$	315150	1600	3h20
BDD Dist.	40		$1.77 \cdot 10^{97}$	20101	725	56.95
	300		$5.39 \cdot 10^{370}$	45151	875	205.45
	500		$4.01 \cdot 10^{622}$	125251	1175	1841.68
	700		$4.89 \cdot 10^{623}$	245351	1482	8 hours



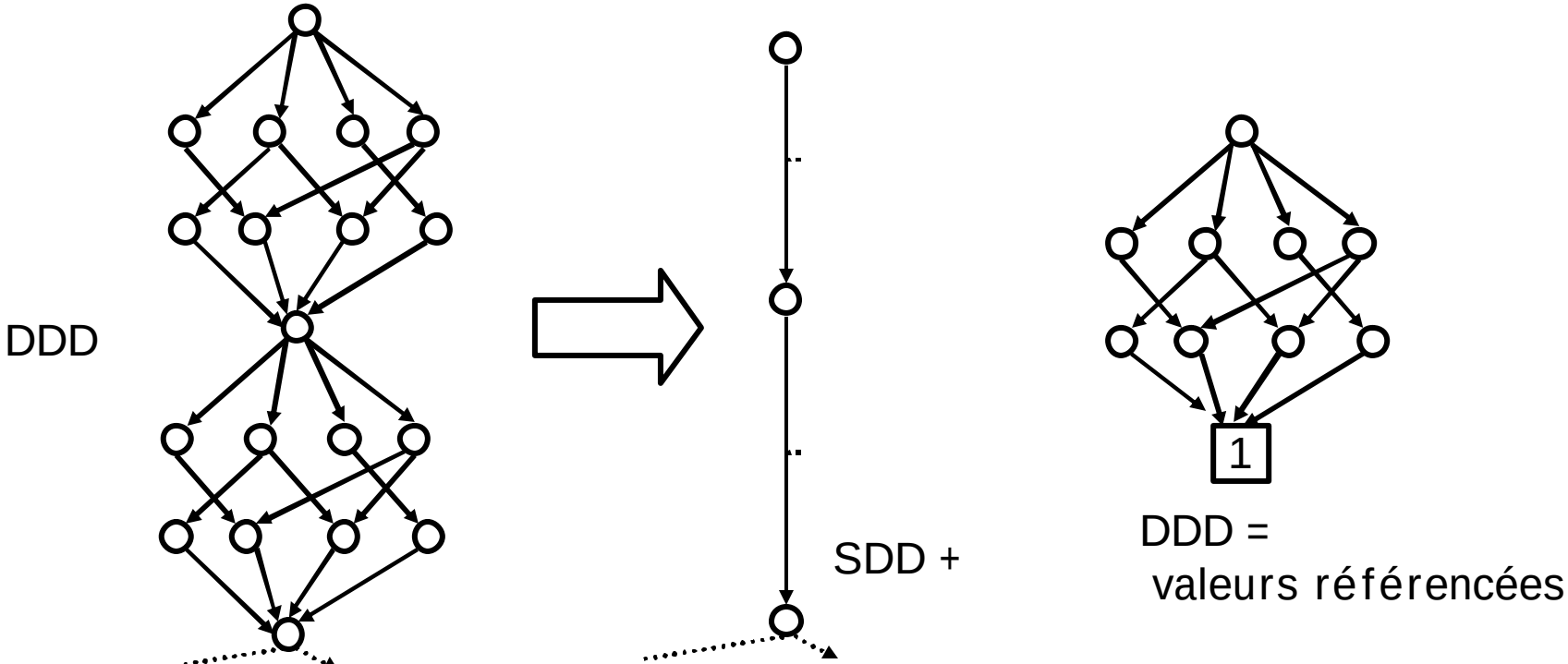
Approches Symboliques et Compositionelles

Méthode et Outils → Analyse des Symétries → Symbolique et Symétries → Symbolique et Hiérarchique

Set Decision Diagram (SDD)

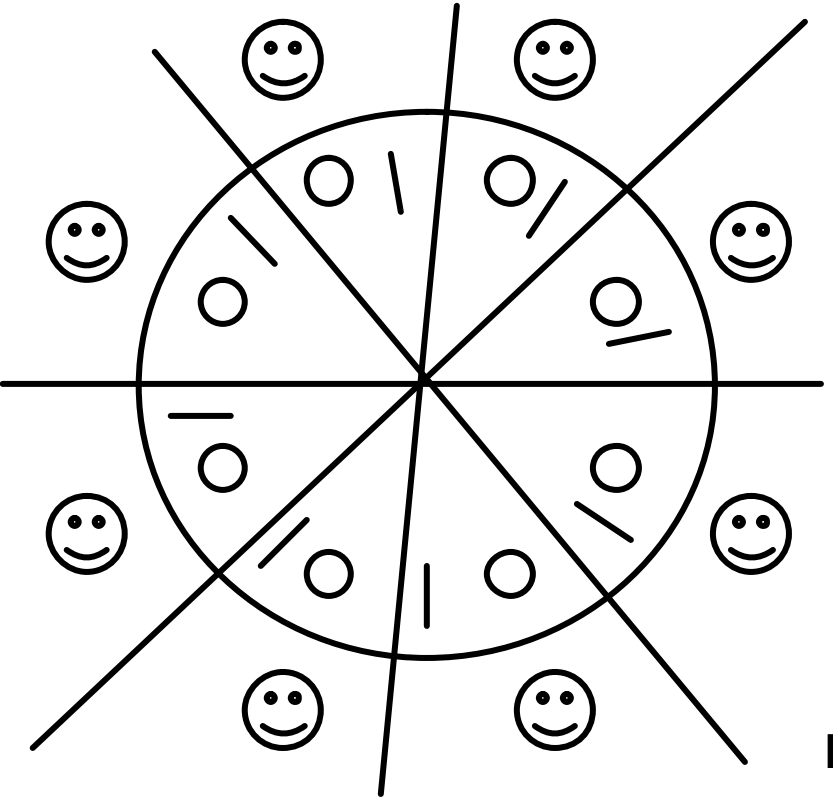
- **Limites de l'existant (DDD) atteintes**
  - **Besoin de structuration**
- **<u>Idée : hiérarchique</u>**
  - **On étiquète les arcs avec des ensembles = Set Decision Diagram**

- **Augmente le partage**
  - **Gain mémoire**
  - **Gain temporel**
    - *cache*
    - *traversée*



Set Decision Diagrams : Modèle Compositionnel

- **Arcs SDD peuvent référencer des DDD**
  - **Structure hiérarchique**
  - **Adapté au compositionnel**
    - *similarité de modules répétés*



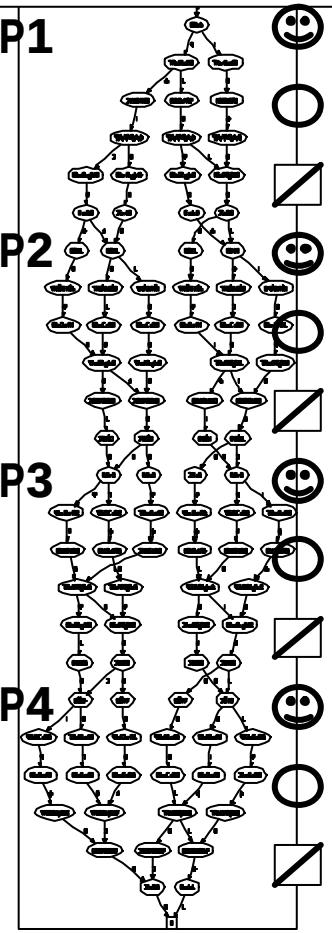
La similarité de comportement entre les philosophes est capturée :

8*(1 Philosophe)

Exemple Philosophes [Dijkstra'65]

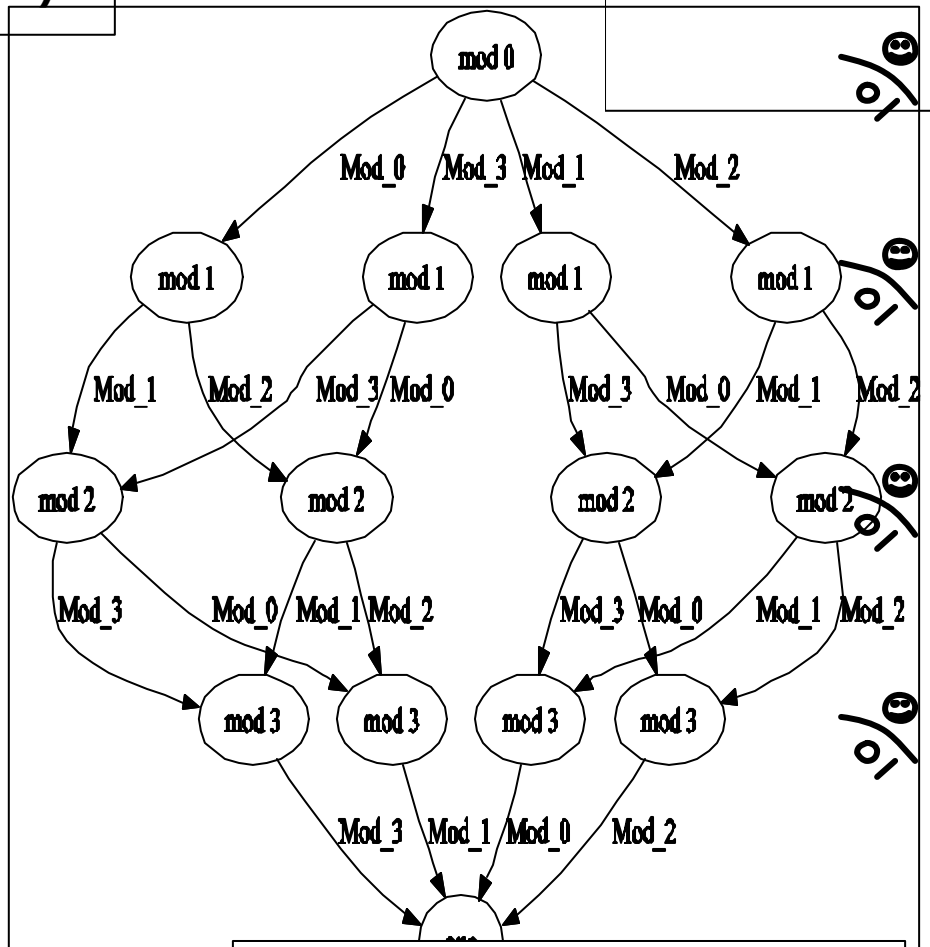
Partage en DDD et Partage en SDD

Espace d'états, 4 philosophes (DDD)

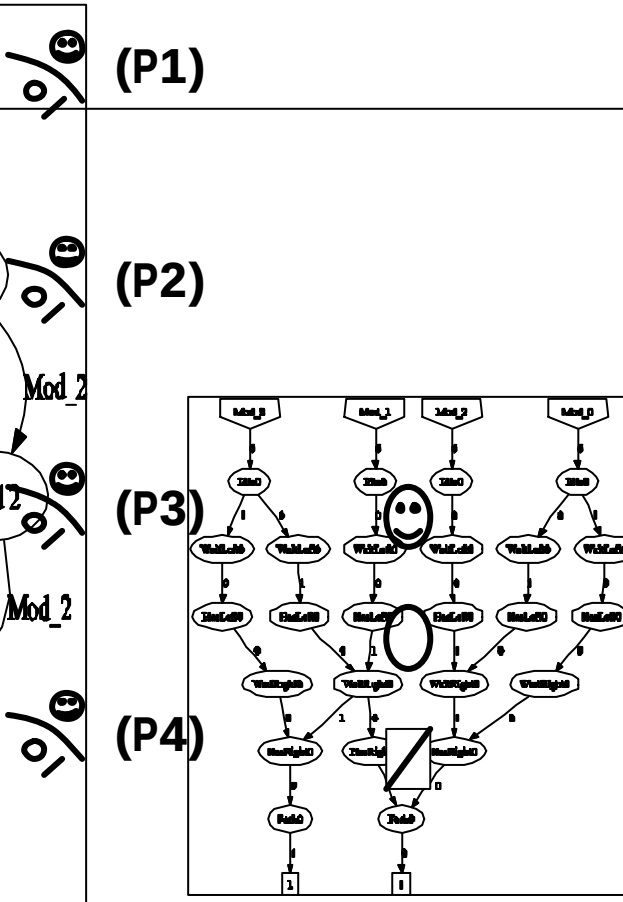


P1 // P2 // P3 // P4

En SDD, l'état d'un philosophe est référencé.



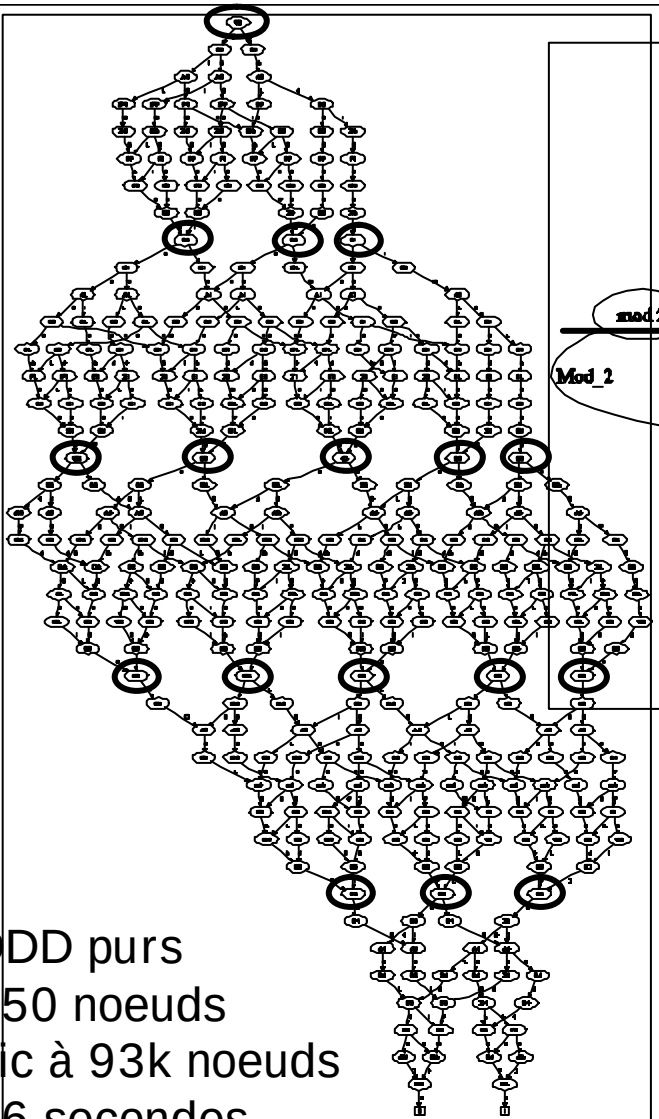
(P1) // (P2) // (P3) // (P4)



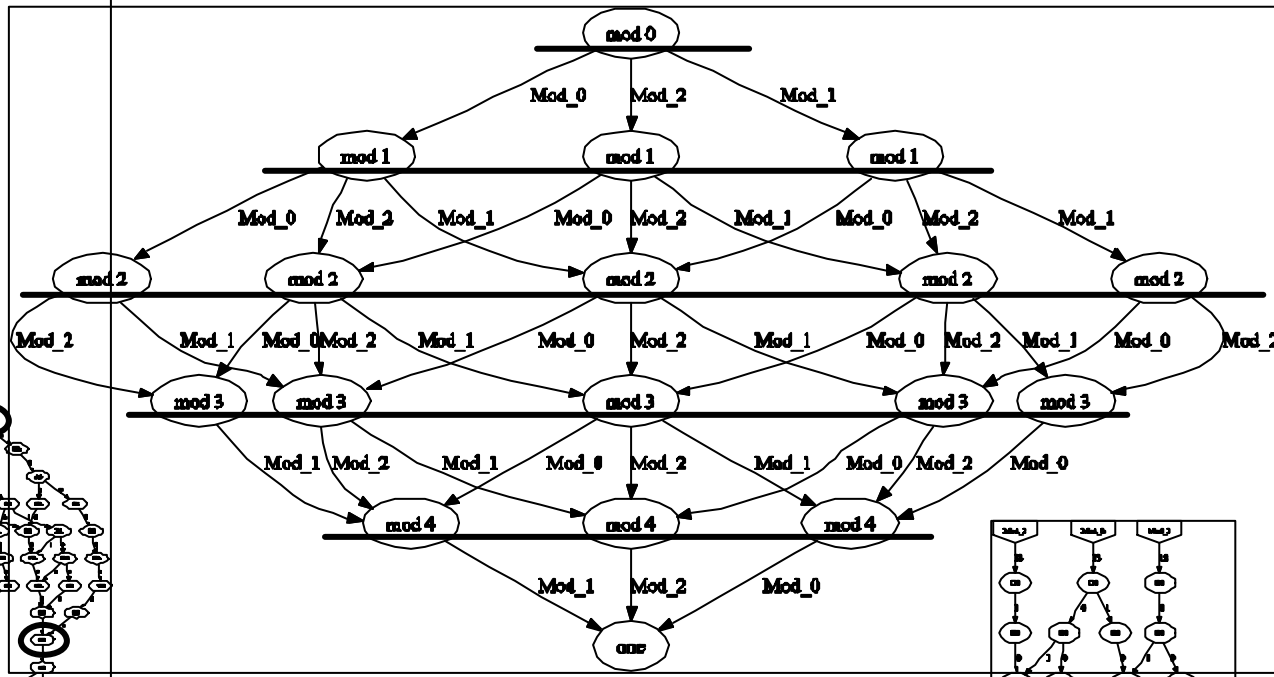
Pi

DDD vers SDD

Modèle "ring", 5 participants, 53856 états

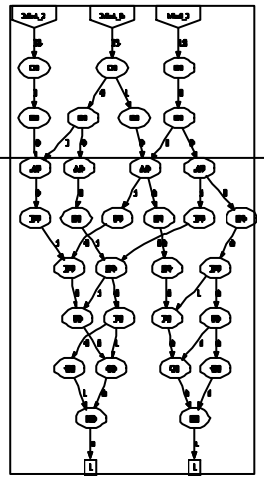


DD purs
50 noeuds
ic à 93k noeuds
6 secondes



SDD ...
18 noeuds

Pic à 1000 noeuds
0.14 secondes



...+DDD
32 noeuds

Symbolique Compositionnel et Saturations Locales

- **SDD bien adaptés au compositionnel**
  - **Les modules indépendants donnent la structuration**
    - *SDD mettent en relation des ensembles d'états*
  - **Calcul de produit synchronisé à la volée**
  - **Gain d'un ordre de grandeur sur les DDD**
- **Localité des actions et point fixe local**
  - **Calcul d'espace d'états = point fixe sur la relation de transitio**
  - **Point fixe décomposés en points fixes locaux [Ciardo'03]**
    - *Ordonner l'évaluation des feuilles vers la racine*
  - **Réduction de la complexité :**
    - *Problème de pic de taille intermédiaire*
    - *Moins d'itérations/ traversées*
  - **Gain d'un ordre de grandeur de plus**

Performances SDD : Prototype PNDDD

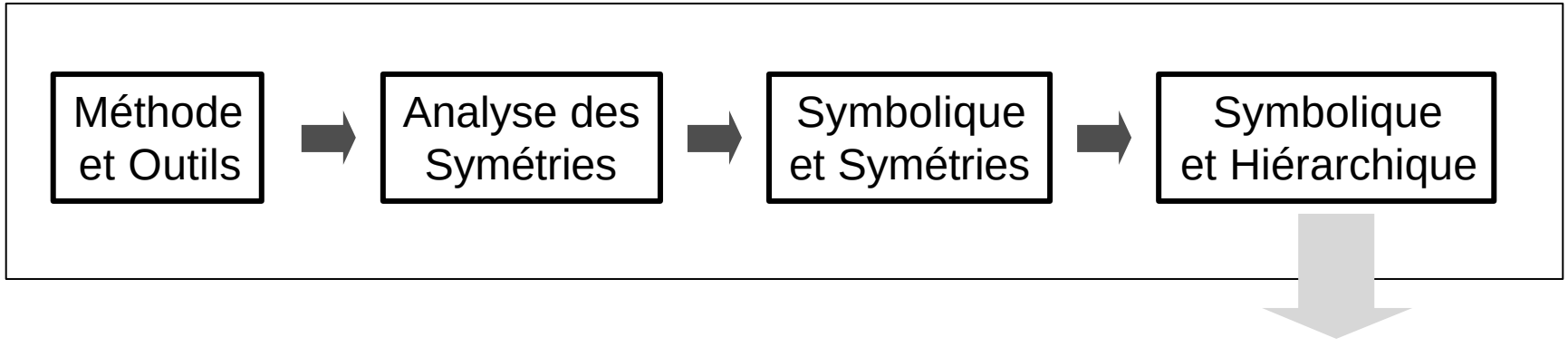
(P4 2.4GHz 2Go) Benchmark Smart

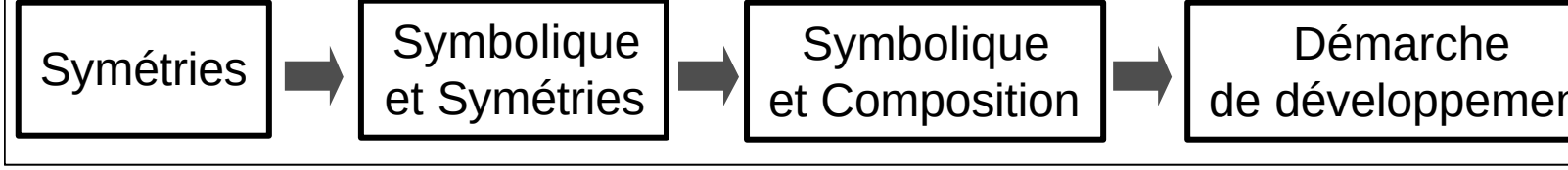
- **Excellente complexité temps et mémoire**
  - **Compare favorablement aux meilleurs outils symboliques disponibles**
  - **Taille intermédiaire gérable**

Model	N (#)	States (#)	Final SDD (#)	Final DDD (#)	PNDDD normal (sec)	PNDDD saturation (sec)	NuSMV temps (sec)	SMaRT temps (sec)
philo	100	4.97+62	398	21	114.58	0.32	990.8	0.43
	200	2.47+125	798	21	-	1.35	18129	0.7
	1000	9.18e+626	3998	21	-	9.03	-	5.9
	5000	6.52+3134	19998	21	-	28.69	-	83.7
kanban	10	1.01e+09	15	46	0.38	0.02	?	0.48
	50	1.04+16	55	206	93.55	0.9	?	43
	100	1.73e+19	105	406	-	6.01	?	474
	200	3.17e+22	205	806	-	48.25	?	13920
FMS	25	8.54e+13	55	412	14.19	0.26	17321	0.36
	50	4.24e+17	105	812	114.63	1.02	-	1.33
	80	1.58e+20	165	1292	-	2.59	-	4
	150	4.8e+23	305	2412	-	10.52	-	20.7
ring	10	8.29e+09	61	44	3.06	0.74	6.1	0.11
	15	1.65e+16	288	44	24.6	2.14	2853	0.29
	50	1.72e+52	2600	44	-	113.14	-	5.6

Offre les gains de la saturation locale dans un framework général

Conclusion

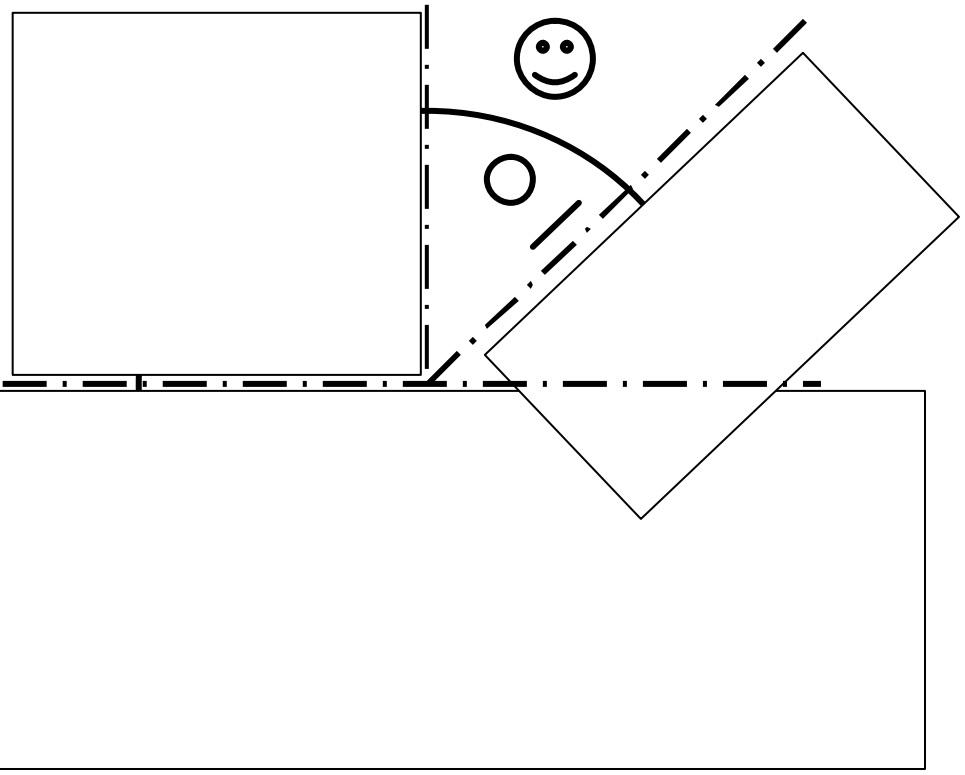

Méthode et Outils
Analyse des Symétries
Symbolique et Symétries
Symbolique et Hiérarchique



- **Problème :**
  - **Vérification de systèmes répartis**
- **Moyens**
  - **Model checking de modèles de haut niveau**
    - *Transformations de modèles*
- **Propositions**
  - **Analyse automatique des symétries (Modèle et propriété)**
  - **Représentations symboliques symboliques (Symétries et Diagrammes de décision)**
  - **Représentations symboliques hiérarchiques (Compositionnel, localité)**
  - **Abstractions Symboliques (Adaptation dynamique à la propriété)**
- **Implémentations (C++, ~20k lignes)**
  - **Set Decision Diagram (LGPL)**
  - **PNDDD (GPL,Réseaux incolores en SDD)**
  - **Snow (CPN-AMI,GreatSPN)**

Pour finir :
Hiérarchie Compositionnelle

- **Arcs SDD peuvent référencer des SDD**
  - **Structure hiérarchique**
  - **Profondeur arbitraire**
- **Exemple Philosophes :**



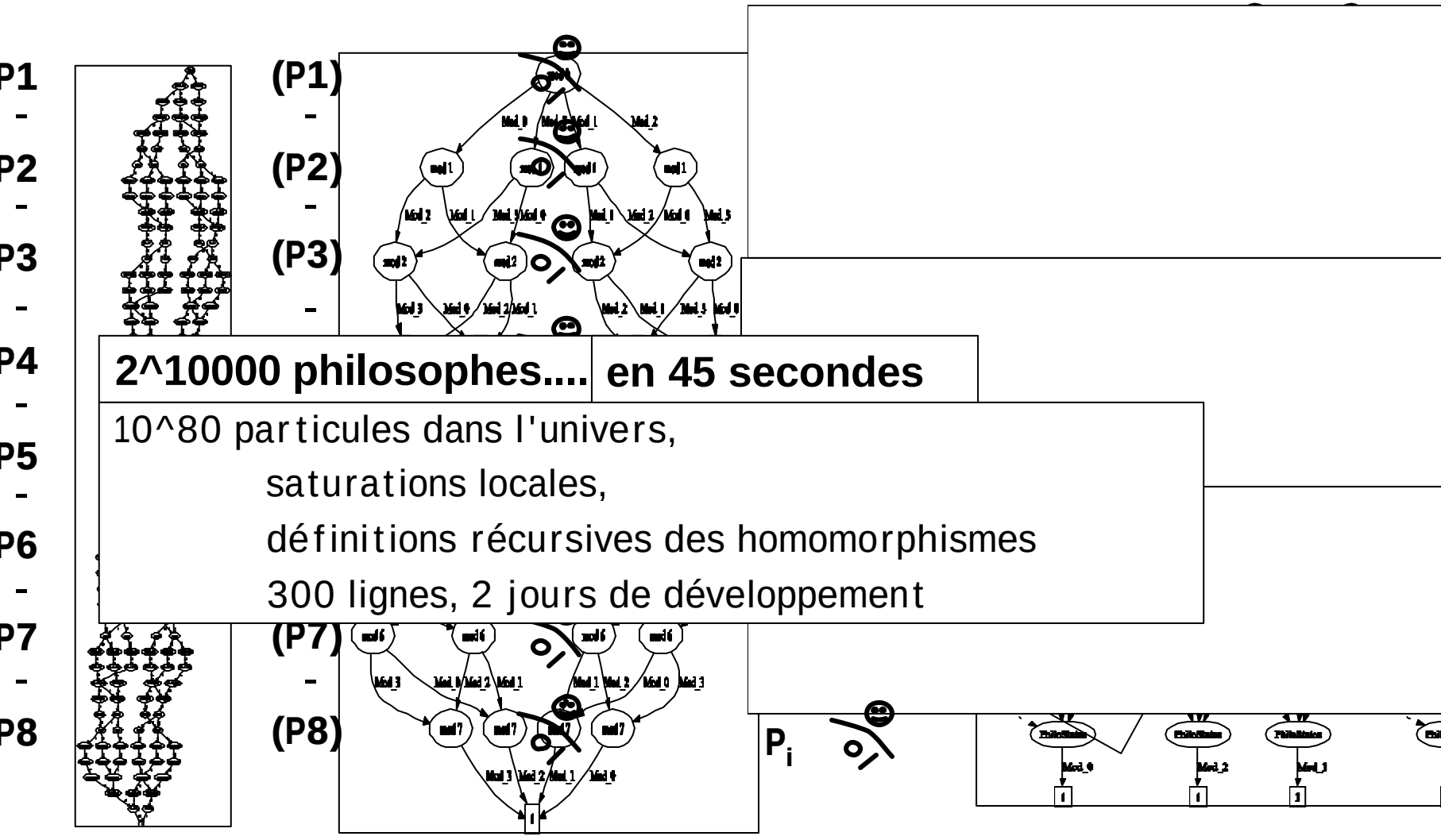
2^3 = 8 philosophes:

3 niveaux de profondeur

+ représentation d'un philosophe

Partage à chaque niveau

Philosophes et Hiérarchie : Potentiel des SDD



Perspectives

- **Set Decision Diagrams**
  - **Symétries et SDD encore à réaliser**
    - *Arcs référencent des ensembles = classe abstraite*
      - Matrix Diagrams, BDD, ...
    - *Adapter la représentation et les opérations aux SDD*
  - **Implémentation des algorithmes LTL symbolique**

  - **Réordonnancement dynamique entre les niveaux**
  - **Généralisation des homomorphisme non-consistents**
  - **Symétries Partielles Symboliques Symboliques**