Junior - Simple

Laurent Hazard
FT R&D – DTL/ASR
16/02/01

Sommaire

- Pourquoi
- Comment
- Exemples
- Une instruction
- Quelques chiffres

Pourquoi ?

- Contexte de recherche, mais…
  - domaine visé
  - inadéquation de l'ordonnancement générique
- Passage à l'échelle
  - des milliers d'objets ?
  - complexité algorithmique
- Efficacité
  - contraintes temps-réel

Passage à l'échelle

- « Scalability »
- Supporter un grand nombre d'activités
  - programmation par objets
  - ce que ne font pas les threads…
  - le critère de mesure de la complexité

Efficacité

- Temps réel
  - dur / mou
  - contrôle + qualité de service
  - notion de contraintes
    - contexte de départ
    - objectifs
  - pas nécessairement « vite »
    - *timeliness*

Anecdotique…

- Complexité des tris
  - intuitifs => N^2
  - optimisés => N log(N)
  - contrainte : ressource « mémoire » limitée
  - difficulté : correction, preuve
  - cas « moyen » / cas « pire »
  - fiabilisé par association ?
    - e.g. : tri rapide + tri bulle

Comment

- Points noirs
  - consommation mémoire
    - nx objets : Rewrite -> Replace
  - déduction des règles de ré-écriture
    - récursivité systématique
    - contexte minimum
  - répétitions de calculs identiques
  - complexité inhérente : $N_{objets} * M_{evts}$

Comment

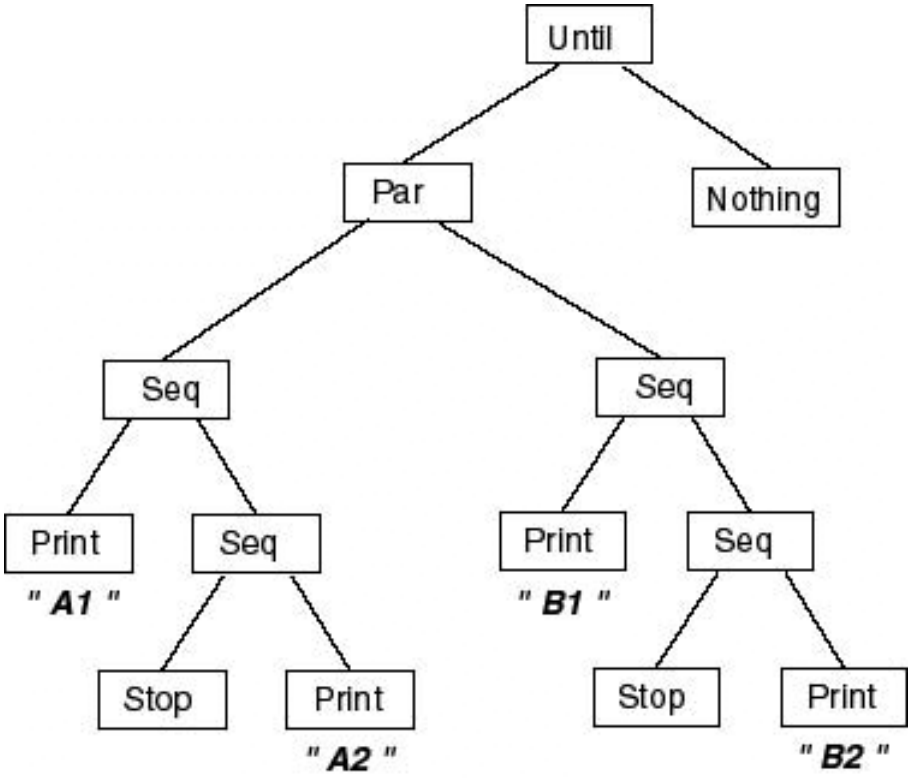
- Principes
  - ne pas refaire…
  - poster les candidats dans une file appropriée
    - (non-)occurrence d'événement
    - exécution systématique
    - étapes intermédiaires
  - complexité min. : N + M
- Difficultés
  - pas de modèle formel de base
  - le contexte devient complexe
  - instruction « parallèle »

Exemple

- ```
  do (
    (print A1; stop; print B1)
  ||
    (print A2; stop; print B2)
  ) until E
  ```

- `instant 1 : A1 A2`
- `instant 2 : B1 B2`
  ```

Exemple


Until
Par
Nothing
Seq
Seq
Print
" A1 "
Seq
Print
" B1 "
Seq
Stop
Print
" A2 "
Stop
Print
" B2 "

Implémentation

- Structures de données
  - `new_pgm` : nouveau programme
  - `reactivable` : instructions ré-activables
  - `event_env` : table des événements
  - `pending` : événements pendants
  - `recall` : instructions filles
  - `flush` : instructions à nettoyer

Table d'événements

- Structure `EventOcc`
  - `name`
  - `fixed`
  - `generated`
  - `sat, both, unsat` : listes d'Awakable
- Table d'`EventOcc`
  - indexée sur le nom
  - nettoyée à la fin de chaque instant

Exécution d'un instant

- Réactivation
- Activation
- Boucle (tant que non vide)
  - réveil des attentes satisfaites
  - exécution des conséquences
- Scrutation de la table des evts
- Terminaison / nettoyage des instructions

Postage d'attente

- Awakable, Awaker
- Presence, NotConfig
- « Descente »

```
satisfiedOrWait (execEnv, eventEnv, aw, wtype, s)
cancelWait (aw, s)
```

  - type d'attente et valeur de retour
    - SAT : PRESENT ou NOTFIXED.
    - UNSAT : ABSENT ou NOTFIXED.
    - BOTH : PRESENT, ABSENT ou NOTFIXED.

Réveil sur occurrence

- « Remontée »
  - Méthode `awake ()`
    - définie sur l'`Awakable` (Instruction ou BinaryConfig)
    - tient compte de la fin d'instant
  - peut être interrompue dans un BinaryConfig

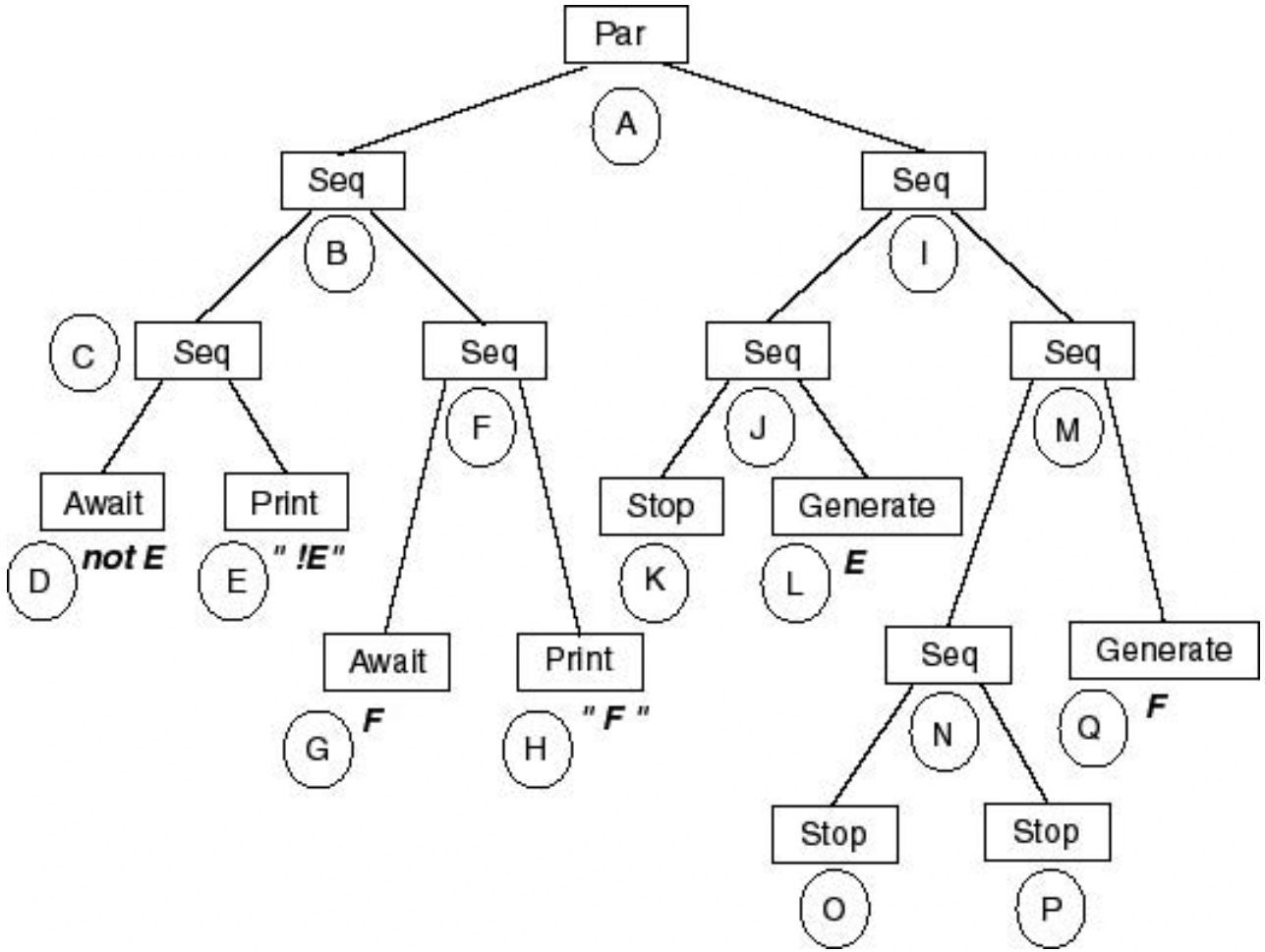
Exemple (simple) - 1

- On considére le programme suivant :

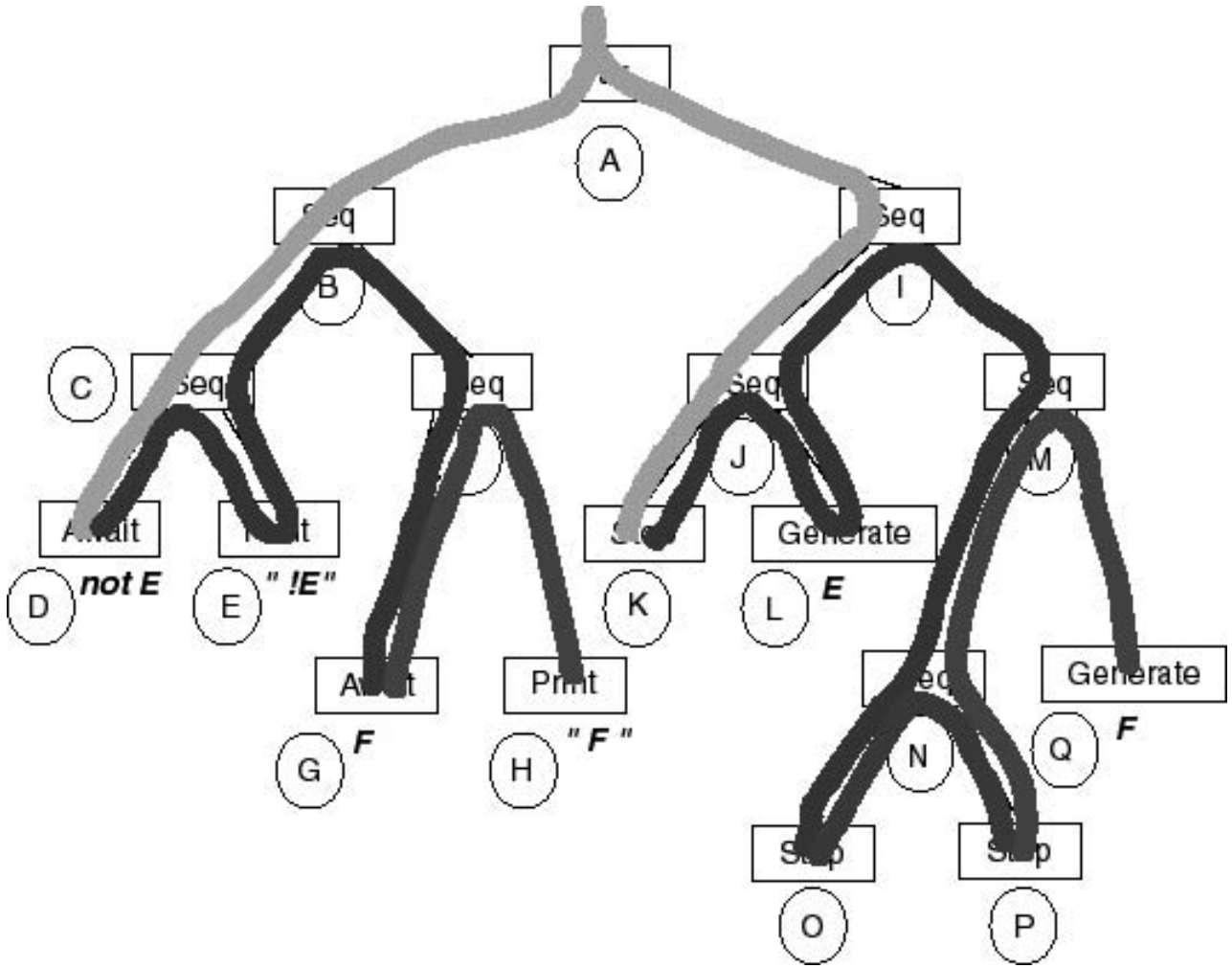
```
await not E; print " !E "; await F; print " F "
||
stop; generate E; stop; stop; generate F
```

- … qui doit donner :
  - `Instant 1:`
  - `Instant 2: !E`
  - `Instant 3:`
  - `Instant 4:  F`

Exemple (simple) - 2



Exemple (simple) - 3



E.g. Until

- Instruction de préemption
- Sémantique (informelle) :
  - exécute le corps tant que la configuration n'est pas satisfaite.
  - exécute le handler si la configuration est satisfaite avant que le corps soit terminé

Quelques chiffres

- Voir comment la simplification algorithmique influe
- Différents exemples correspondant à :
  - des situations plausibles (?)
  - des gains (?) de nature différentes
- D'autres critères intéressants
  - pile
  - table d'événements
  - instructions terminées

Bilan

- efficace, mais pas facile…
- probablement bogué

carence d'un modèle -plus- théorique

- gain pas uniforme (pas systématique)
- compromis…
- questions ?