– reROUTEable –

June 29, 2006

http://www.cs.uni-salzburg.at/~ck/wiki/index.php?n=TCS-Summer-2006.ReRouteable

- Introduction
- Syntactical Definitions
- Semantics
- Algorithms
- Conclusion
- Further Work

- reMOTEable
- aims
  - formalization of the agent system reMOTEable
  - focus on enhancing the command-set (by reduction)
  - the computation model is a formalized Virtual Machine
  - proving by applying algorithms on the model

Syntactical Definitions

A network Nw (N, C) is a 2-tuple, where:

- **N** is the set of all nodes
- C are the connections: $C := \{\{a, b\} | a, b \in \mathbf{N} \wedge a \text{ is connected with } b\}$

An Agent is an 4-tuple (PGM, PC, MEM, CNODE), where:

- PGM is the program of the agent.
- PC is the programcounter: PC $\in \mathbb{N}$.
- MEM is the memory of the agent.
- CNODE $\in \mathbf{N}$: the node on which the agent is currently residing. Note an agent can only be active in exactly one node at a certain time: $\exists! n \in \mathbf{N} : \forall a \in \mathbf{A}$.

A is the set of all agents.

- A program PGM is a n-tuple:
  $CMD^n = \underbrace{CMD \times CMD \times CMD \cdots \times CMD}_{n-times}$
- The memory MEM consists of three parts: pathMem, tempMem, targetMem.
  Where all parts are n-tuples of nodes.
- $\bigstar(x) := \{y \in \mathbf{N} | \{x, y \in \mathbf{N}\} \wedge \in \mathbf{C}\}$
  Predicate $visited(a, n)$ is true if the agent a, has visited n.

Possible instruction sets:

- $I_1 := \{$ PUSH, POP, MOVE, CMPJMP, LOADNEXT $\}$ without replication.
- $I_2 := \{$ PUSH, POP, MOVE, CMPJMP, MOVEREPLICATE $\}$ for flooding.

- Formalized Virtual Machine
- Access function for tuples: $G_n(x_1, x_2, \ldots, x_{n-1}, x_n) := x_n$ G stands for grab.

$$PUSH(A, B) = \{ \quad B : (b_1, b_2, \ldots, b_n) \longmapsto B : (b_1, b_2, \ldots, b_n, G_n(A))$$

$$POP(B) = \{ \quad B : (b_1, b_2, \ldots, b_n) \longmapsto B : (b_1, b_2, \ldots, b_{n-1})$$

$$CMPJMP(A, B, c) = \{ \; G_n(A) = G_n(B), \quad PC \longmapsto c$$

$$MOVE(B) = \begin{cases} B = \emptyset, & \mathbf{A} \longmapsto \mathbf{A} \setminus \{a_{this}\} \\ \text{otherwise}, & CNODE \longmapsto G_n(B) \end{cases}$$

LN ... LOADNEXT; MR ... MOVEREPLICATE

$$LN(B) = \begin{cases} \exists x \in \bigstar(CNODE) \wedge \neg visited(x), & B : (b_1, \ldots, b_n) \longmapsto B : (b_1, \ldots, b_n, x) \\ \text{otherwise}, & B : (b_1, \ldots, b_n) \longmapsto B : (b_1, \ldots, b_{n-1}) \end{cases}$$

$$MR(B) = \begin{cases} \forall x \in \bigstar(CNODE) \wedge \neg visited(x) : \text{generate } a_{new} \in \mathbf{A} \\ \text{with } (a_{old}.PGM, a_{old}.PC, a_{old}.MEM, a_{old}.x) \\ \mathbf{A} \setminus a_{old} \end{cases}$$

```
010 PUSH target, targetMem
020 PUSH currentNode, pathMem
030 CMPJMP currentNode, targetMem, 80
040 LOADNEXT pathMem
050 MOVE pathMem
060 CMPJMP 0,0,30
070 PUSH currentNode, tempMem
080 POP pathMem
090 MOVE pathMem
100 CMPJMP 0,0,10
```

```
010 PUSH target, targetMem
020 PUSH currentNode, pathMem
030 CMPJMP currentNode, targetMem, 60
040 MOVEREPLICATE pathMem
050 CMPJMP 0,0,20
060 PUSH currentNode, tempMem
070 POP pathMem
080 MOVE pathMem
090 CMPJMP 0,0,10
```

- computational model works for two algorithms
- 5 instructions sufficient for routing (compared 15 instructions in Remoteable)
- two complex instructions requiring logic in Node (one for each algorithm)

- is there a simpler model (trade off Memory Complexity vs additonal Commands)?
- express LoadNext with PUSH, POP, MOVE, CMPJUMP
- evaluate extendability of routing model to all functionalities of ReMOTEable