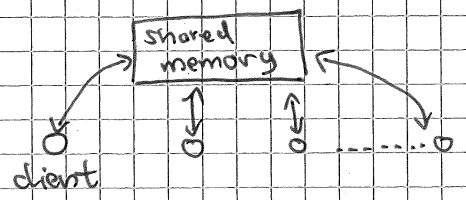
Broadcast Networks

Goal: Verification of parameterized systems:



→ Identical clients communicating via shared memory
→ Arbitrary number of clients!

Applications: Cache-coherence protocols,
distributed algorithms - leader election.
⋮

Verification difficult: Nr. of clients can be arbitrary.

We look at broadcast networks - special class of parameterized systems. Clients send messages to each other.

We show: • Safety (Reachability) is in P (2012)
• Liveness (Repeated Reachability) is in P (2019)

Definition: Broadcast network

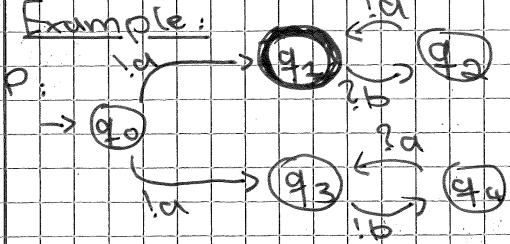
A broadcast network is a pair $\mathcal{N} = (D, P)$, with

- data domain D, a finite set of messages,
- client $P = (Q, I, \delta)$, an NFA over
- the operations $Ops(D) = \{!a, ?a \mid a \in D\}$.
  ($!a$: send a; $?a$: receive a)

with $\delta \subseteq Q \times Ops(D) \times Q$ and $I \subseteq Q$ a set of init. states.

In a broadcast network $\mathcal{N}$, the nr. of clients is arbitrary. But each client is identical to P.

Example:



Let us assume, there are two clients - two copies of P: P_1 and P_2.

A configuration should keep track of both copies:

(q, q')
q: state of P_1; q': state of P_2

If we have ~~...~~ $P_1, \dots, P_n$, a config. should be $(q_{i_1}^1, \dots, q^n)$, holding the current state of each P_i.

Definition: Configurations

The set of configurations is $CT = \bigcup_{k \in \mathbb{N}} Q^k$.

A configuration is thus a tuple $(q_{i_1}, \dots, q_{i_k})$, where k is the nr. of clients involved in the computation. The set of initial configurations is $CT_0 = \bigcup_{k \in \mathbb{N}} I^k$.

Definition: Semantics

Let $c = (q_{i_1}, \dots, q_{i_k})$ and $c' = (q'_{i_1}, \dots, q'_{i_k})$ configurations and Let $a \in \Theta$.

We have $c \xrightarrow{a}_K c'$ if the following holds:

(1) $\exists i \in \{1, \dots, k\}$ with $q_i \xrightarrow{!a} q'_i$ // there is a sender

(2) $\exists R \subseteq \{1, \dots, k\} \setminus \{i\}$ with $q_j \xrightarrow{?a} q'_j \ \forall j \in R$

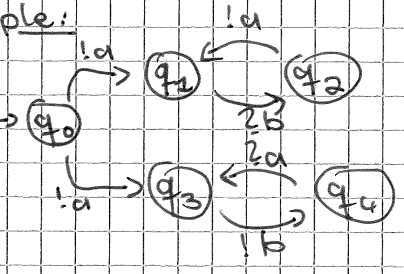
// there are receivers.

(3) $\forall j \notin R \cup \{i\}$ we have $q_{i_j} = q'_j$ // other clients stay idle.

Remarks:

- Configurations in a computation have the same nr. of clients.
- It is not possible to "create" clients during comp. (Dynamic thread creation).

Example:



Computation with two clients:

$(q_0, q_0) \xrightarrow{a} (q_1, q_0) \xrightarrow{a} (q_2, q_3) \xrightarrow{b} (q_2, q_4) \xrightarrow{a} (q_1, q_3)$.

P_1 sends, no one receives

P_2 sends, no one receives

P_2 sends, P_1 receives

③ Safety Verification:

Safety $\hat{=}$ can program/system reach an unsafe state?

→ A question of reachability.

→ Solve the following problem:

BN Reachability:

Input: A broadcast network $\mathcal{N} = (D, P)$, a set of final states $F \subseteq Q$.

Question: Is there a computation $C_0 \xrightarrow{*}_{\mathcal{N}} C$ with $C \in C_{F_0}$ and $Set(C) \cap F \neq \emptyset$?

→ If $C = (q_{i_1}, \dots, q_{i_k})$, $Set(C) = \{q_{i_1}, \dots, q_{i_k}\}$.

→ BN Reachability asks for at least one client to reach a state in F.

Problem: How to test reachability of such a configuration?

→ C_F is infinite! We do not know the number of involved clients.

Approach: Use this as an advantage!
We can assume to have arbitrarily many clients whenever we need them.

Example: Consider P from the earlier example.

There is a computation $\xi: (q_0, q_0) \xrightarrow{a} (q_1, q_0) \xrightarrow{a} (q_2, q_0) \xrightarrow{b} (q_3, q_4)$.

We can use more clients so that we do not loose states. Computation simulates ξ:

We loose the state q_0 here.
→ client in q_0 moves to q_3.

$\xi': (q_0, q_0, q_0) \xrightarrow{a} (q_1, q_0, q_0) \xrightarrow{a} (q_2, q_3, q_0) \xrightarrow{b} (q_3, q_4, q_0)$

we loose q_2, q_3.

Add further clients that mimic behavior of first and second client and stop in q_2, q_3:

$\xi'': (q_0, \underline{q_0}, q_0, \underline{q_0}, q_0) \xrightarrow{a} (q_1, \underline{q_0}, q_0, \underline{q_0}, q_0) \xrightarrow{a} (q_2, \underline{q_2}, q_0, \underline{q_0}, q_0)$

↓ copy of first

↓ copy of second

↓ copy mimics more of first client

$\xrightarrow{a}(q_2, \underline{q_1}, q_3, \underline{q_0}, q_0) \xrightarrow{a} (q_1, \underline{q_1}, q_3, \underline{q_3}, q_0)$

↓ now they stop

$\xrightarrow{b}(q_2, q_1, q_4, q_3, q_0)$. Now we have a client in each state.

$\Rightarrow$ We can add clients to computations so that we do not loose states anymore.

AND: We can assume the nr. of clients in a particular state to be as large as we need it. (By adding enough copies).

Remarks:

Since we can assume that in a computation, there are arbitrarily many clients in a particular state, knowing the precise configuration is not important.

→ We only need to know the set of states used.

→ The following graph abstracts from configurations to sets:

Definition: Abstraction graph

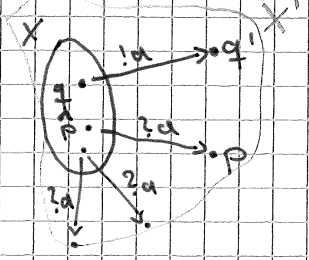
Let $\mathcal{N} = (P, P)$ be a B.N. with $P = (Q, I, \delta)$.

The abstraction graph of $\mathcal{N}$ is a graph $\mathcal{G} = (\mathcal{P}(Q), E)$ with edges defined as follows:

$X \xrightarrow{}_E X'$ if $\exists a \in \Sigma$ and a $q \in X$ with a transition $q \xrightarrow{!a} q'$ s.th.

$$X' = X \cup \underbrace{\{q'\}}_{\text{target of sender}} \cup \underbrace{\{p \in Q \mid \exists \hat{p} \xrightarrow{?a} p, \hat{p} \in X\}}_{\text{targets of receivers}}.$$

→ An edge simulates a transition of $\mathcal{N}$. There is a sender moving from $q \xrightarrow{!a} q'$ and receivers.



(5)

Lemma: Let $\mathcal{N}=(Q,P)$ be a B.N. and $F \subseteq Q$.

There is a computation $c_0 \to_{\mathcal{N}}^* c : Set(c) \cap F = \emptyset$ $\iff$ A state X of $abs(P)$ with $X \cap F \neq \emptyset$ is reachable from I.

Proof:

"$\Rightarrow$": Let $\rho = c_0 \to_{\mathcal{N}} c_1 \to_{\mathcal{N}} c_2 \to \dots \to_{\mathcal{N}} c_\ell = c$ be a comp. in $\mathcal{N}$ with $Set(c) \cap F \neq \emptyset$.

We show that there is a path $\pi = Q_0 \to_E Q_1 \to_E \dots \to_E Q_\ell$ in $abs(P)$ s.th. $I = Q_0$ and $\forall i \in \{1,\dots,\ell\}$ we have: $Set(c_i) \subseteq Q_i$.

→ Note: this shows that $Q_\ell \cap F \neq \emptyset$.

→ Construct π by induction:

I.B.: $i=0$ (init. conf. $\subseteq I = Q_0$)

Set $Q_0 := I$. Then $Set(c_0) \subseteq I = Q_0$.

I.S.: Assume we constructed $Q_0 \to_E \dots \to_E Q_i$.

Consider (i+1)-th transition of ρ: $c_i \xrightarrow{a}_{\mathcal{N}} c_{i+1}$.

$\overset{Def.}{\Rightarrow}$ for $c_i = (q_1,\dots,q_n)$, $c_{i+1} = (q_1',\dots,q_n')$

(1) $\exists j \in \{1,\dots,k\} : q_j \xrightarrow{!a} q_j'$,

(2) $\exists R \subseteq \{1,\dots,k\} \setminus \{j\} : q_h \xrightarrow{?a} q_h' \ \forall h \in R$

(3) $\forall h \notin R \cup \{j\} : q_h' = q_h$.

Since $Set(c_i) \subseteq Q_i \Rightarrow q_j \in Q_i$. So we can mimic the transition with an edge in $abs(P)$:

Set $Q_{i+1} := Q_i \cup \{q_j'\} \cup \{p \in Q \mid \exists \hat{p} \xrightarrow{?a} p, \hat{p} \in Q_i\}$. (the latter set $=: S$)

Then, we get an edge $Q_i \to_E Q_{i+1}$.

Moreover, we have $Set(c_{i+1}) \subseteq Q_{i+1}$:

(1) $q_j' \in Q_{i+1}$

(2) $\forall h \in R \Rightarrow q_h \in Q_i$ and hence $q_h' \in S \subseteq Q_{i+1}$

(3) $\forall h \notin R \cup \{j\} \Rightarrow q_h' = q_h \in Q_i \subseteq Q_{i+1}$.

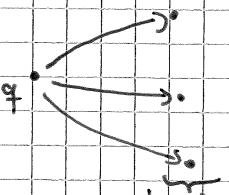
"⇐": Let $\pi = Q_0 \to Q_1 \to \dots \to Q_\ell$ be a path in abs(P) with $Q_0 \overset{\text{init}}{\cap} I$, $Q_\ell \cap F \neq \emptyset$.

We construct a computation in $\mathcal{N}$:

$$\xi = c_0 \to^*_{\mathcal{N}} c_1 \to^*_{\mathcal{N}} \dots \to^*_{\mathcal{N}} c_\ell \text{ with } Set(c_\ell) \cap F \neq \emptyset.$$

→ How many clients should we use in ξ?

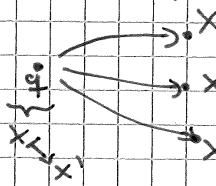
Idea:



In a transition of $\mathcal{N}$, [redacted] clients in state q change their state to one in post(q).

We construct the computation ⇒ we choose the nr. of clients that move to another state.

Assume we have X clients in q. Then we send X' to each state in post(q) and we want to keep X' in q.



$$\Rightarrow X \geq \underbrace{X'}_{\text{clients remaining in } q} + \underbrace{(|Q|-1)}_{|post(q) \setminus \{q\}|} \cdot X' \geq X' \cdot |Q|.$$

The path π has length ℓ, and ξ should also have that length.

⇒ X_i := nr. of clients in state q after i steps:

$$X_0 \geq X_1 \cdot |Q| \geq X_2 \cdot |Q|^2 \geq \dots \geq \underbrace{X_\ell}_{\geq 1} \cdot |Q|^\ell$$

// Having one client in q at the end is enough

$$\Rightarrow X_0 \geq |Q|^\ell.$$

→ We construct ξ inductively over Q^n, where $n = |Q|^\ell$.

Moreover, ξ satisfies: (1) $Set(c_i) = Q_i$

(2) $\underbrace{\#(c_i, q)}_{\text{nr. of clients in state } q, \text{ in } c_i} \geq |Q|^{\ell - i} \quad \forall q \in Q_i$.

(7)

I.B: $i=0$: w.l.o.g. $I=\{q_0\}$.

$c_0 := (\underbrace{q_0, \dots, q_0}_{n\text{-times}})$

$\Rightarrow$ (1): $\text{Set}(c_0) = \{q_0\} = I = Q_0$.

(2): $\#(c_0, q_0) = n = |Q|^\ell$.

I.S: Assume, we constructed $c_0 \to_{\mathcal{N}} \dots \to_{\mathcal{N}} c_i$ with (1), (2).

Consider $Q_i \to_\ell Q_{i+2}$ in $\mathcal{T}$.

$\overset{\text{def.}}{\Rightarrow}$ $\exists\, a \in \Sigma$ and $q \in Q_i$: $q \xrightarrow{!a} q'$ and

$Q_{i+2} = Q_i \cup \{q'\} \cup \{p \in Q \mid \exists\, \hat{p} \xrightarrow{?a} p,\ \hat{p} \in Q_i\}$.

We simulate the edge by a sequence of transitions in $\mathcal{N}$.

- First, we do send transitions: $q \in Q_i \overset{(1)}{=} \text{Set}(c_i)$. Construct

$c_i \xrightarrow{a}_{\mathcal{N}} c_i^{(2)} \xrightarrow{a}_{\mathcal{N}} \dots \xrightarrow{a}_{\mathcal{N}} c_i^{(|Q|^{\ell-(i+2)}-1)} =: \hat{c}_i$ by appending the following transition $|Q|^{\ell-(i+2)}-1$ many times:

some client in q does $q \xrightarrow{!a} q'$, no client receives.

$\overset{(2)}{\Rightarrow}$ There are $\#(c_i, q) \geq |Q|^{\ell-i} \geq |Q|^{\ell-(i+2)}$ many clients in q, so this is possible to construct.

- Then, we construct $\hat{c}_i \xrightarrow{a}_{\mathcal{N}} c_{i+2}$ by:
  - Let one client in q do the send $q \xrightarrow{!a} q'$,
  - Receivers are: for each $\hat{p} \in Q_i$ with $\hat{p} \xrightarrow{?a} p$, let $|Q|^{\ell-(i+2)}$ many clients in $\hat{p}$ move to p. (There are enough (2)).

$\Rightarrow$ we obtain c_{i+2} and it satisfies (1) and (2):

(1): $\text{Set}(c_{i+2}) = Q_{i+2}$: $q' \in \text{Set}(c_{i+2})$, $\{p \in Q \mid \exists\, \hat{p} \xrightarrow{?a} p, \hat{p} \in Q_i\} \subseteq \text{Set}(c_{i+2})$, and we do not loose states $\Rightarrow$

$Q_i = \text{Set}(c_i) \subseteq \text{Set}(c_{i+2})$.

(2): Let $s \in Q_{i+1}$.

~~Case 1: q_f belongs to a di~~

Case 1: $s = q_f'$ or $s = p'$, $p' \in \{ p \in Q \mid \exists \hat{p} \xrightarrow{?a} p, \hat{p} \in Q_i \}$

By construction, there are $|Q|^{l-(i+2)}$ many clients that move to state s (senders or receivers).

Case 2: $s \in Q_i$

By construction, the clients in s move to at most $|Q|-1$ many new states. Each time, we let $|Q|^{l-(i+2)}$ many move.

$\Rightarrow$ Nr. of clients remaining in s is

$$\#(c_{i+1}, s) \geq \underbrace{\#(c_i, s)}_{\text{clients in } s \text{ in } c_i} - \underbrace{|Q|^{l-(i+2)}}_{\substack{\text{nr. of clients we} \\ \text{move to} \\ \text{a post-state}}} \cdot \underbrace{(|Q|-1)}_{\substack{\text{post-states} \\ \text{of } s}}$$

$$\overset{(2)}{\geq} |Q|^{l-i} - \left(|Q|^{l-i} - |Q|^{l-(i+1)} \right) = |Q|^{l-(i+2)}. \quad \blacksquare$$

Remarks:

- It is enough to find a path in $abs(P)$
- But $abs(P)$ is of exponential size.
- Paths are "increasing" $\rightarrow$ can apply a saturation.

Definition: Post operator

Let $X \in \mathcal{P}(Q)$, we define

- $post(X) := \{ s' \in Q \mid \exists s \in X : s \xrightarrow{!a} s' \}$
$$\cup \{ s' \in Q \mid \exists s_1, \underset{s_2 \in Q}{s \in X} : s_1 \xrightarrow{!a} s_2 \wedge s \xrightarrow{?a} s' \}$$
- $post^*(X) = \bigcup_{k \in \mathbb{N}} post^k(X) \in \mathcal{P}(Q)$.

(9)

Lemma: Properties of post

a) For $X \in \mathcal{P}(Q) \Rightarrow X \to_E^* Y \supseteq post(X) \cup X$

b) If $X \to_E^* X' \Rightarrow X' \subseteq post(X) \cup X$.

Proof: Exercise.

With the properties, we get the correctness of our later algorithm:

Lemma: Correctness criterion

$post^*(I) \cap F \neq \emptyset$ $\Leftrightarrow$ there is a path in $abs(P)$, leading from I to X: $X \cap F \neq \emptyset$.

Proof:

„$\Rightarrow$": $post^*(I) = \bigcup_{k \in \mathbb{N}} post^k(I) = \bigcup_{k=1}^{n} post^k(I)$

($\mathcal{P}(Q)$ is finite)

Apply a) from above inductively.

„$\Leftarrow$": Let $\pi := \underbrace{I}_{=:X_0} \to X_1 \to \dots \to X_\ell = X$ be the path.

$\overset{b)}{\Rightarrow}$ by induction $X_i \subseteq post^i(I) \cup post^{i-1}(I) \cup \dots \cup I$

$\Rightarrow X \subseteq post^*(I)$. ∎

So it is left to compute the set $post^*(I)$.
We can do this with a fixed-point iteration:

Lemma: Fixed-point

Let $X \in \mathcal{P}(Q)$, then $post^*(X)$ can be computed in polynomial time.

Proof:

Given X, $post(X)$ can be computed in $\mathcal{O}(|\delta|^2)$ time.

To compute $post^*(X)$, apply saturation:

Initialize $R := X$, $oldR := \emptyset$.

```
while (R ≠ oldR) {
    R := R ∪ post(R);
    oldR := R;
}
return R;
```

R can grow for at most $|Q|$ iterations since $R \subseteq Q$.

When $R = oldR \Rightarrow R = post^*(X)$.

$\Rightarrow \mathcal{O}(|Q| \cdot |\delta|^2)$ time. ∎